

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА


Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

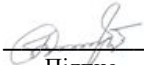
на тему «Метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF»

ДРКІ 190180.22.01.10 ПЗ

Виконав: студент 2 курсу, група КІ2М-22-1

 Б.В. Глімбовський
Підпис Ініціали, прізвище

Керівник доктор техн. наук, професор
Науковий ступінь, вчене звання

 О.В.Боровик
Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КІС, д.т.н., проф.
Т.О. Говорущенко

21 травня 2024 р.

Хмельницький, 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма освітньо-наукова програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 01 ” 09 2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Глімбовському Богдану Валентиновичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF

Керівник проекту (роботи) Боровик О.В., д.т.н., професор

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.01.2024 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Опис предметної області

Розробка методу та дослідження ефективності

Вимоги до системи та технології розробки

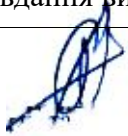
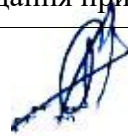


Дослідження ефективності та тестування системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Макети мобільного додатку _____

Зображення у мобільному додатку _____

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	01.09.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2023	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2023	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2023	виконано
5	Робота над науковою статтею	01.02.2024	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2024	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2024	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2024	виконано
9	Попередній захист ДРМ	29.04.2024	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2024	

Студент


Підпис

Глімбовський Б.В.

Ініціали, прізвище

Керівник роботи


Підпис

Боровик О. В.

Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод опису метаданих об'єктів для мобільної кроссплатформної CRM-системи на платформі VAF

Автор роботи: Глімбовський Богдан Валентинович

Керівник роботи: Боровик Олег Васильович

Пояснювальна записка: 81 с., 56 рис., 22 табл., 5 дод., 81 джерел.

МОБІЛЬНА CRM СИСТЕМА, МЕТАДАНИ, КРОСПЛАТФОРМЕННІСТЬ, VAF ПЛАТФОРМА, HTTP СЕРВІСИ, ПРОТОКОЛ ODATA, БЕЗПЕКА ДАНИХ, ЕФЕКТИВНІСТЬ ОБМІНУ ДАНИМИ, XAMARIN, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – CRM система, інформаційні технології, моделі опису метаданих об'єктів у CRM-системах.

Предмет дослідження - метод опису метаданих об'єктів для мобільної кроссплатформної CRM-системи на платформі VAF.

Мета кваліфікаційної роботи магістра – дослідити та розробити ефективний метод опису метаданих об'єктів для мобільної CRM системи на платформі VAF, забезпечивши оптимальну швидкодію, безпеку та зручність використання. За результатом виконання роботи були поставлені та вирішені наступні завдання:

1. Аналіз існуючих методів опису метаданих: Проведення огляду наукової літератури та аналіз існуючих підходів до опису метаданих в мобільних CRM системах для визначення переваг та недоліків існуючих методів.

2. Розробка нового методу опису метаданих: На основі отриманих даних та виявлених проблем розроблено новий метод опису метаданих, який має на меті покращити ефективність обміну даними та забезпечити більшу гнучкість та масштабованість системи.

3. Впровадження та тестування нового методу: Реалізація розробленого методу в мобільній CRM системі та проведення його тестування для перевірки функціональності та ефективності в реальних умовах використання.

4. Оцінка результатів та висновки. Аналіз отриманих результатів тестування нового методу та формулювання висновків щодо його ефективності та можливостей подальшого вдосконалення.

Для досягнення поставлених задач виконано аналіз існуючих рішень, сформовано функціональні вимоги до системи, спроектовано та розгорнуто систему у хмарній платформі з можливістю доступу через мережу Інтернет.

Наукова новизна отриманих результатів полягає у розробці нового методу опису метаданих об'єктів, який сприяє більш ефективній роботі з даними, та створенні мобільного додатку, який реалізує зазначений метод.

Практична значимість отриманих результатів полягає в забезпеченні можливостей для удосконалення механізмів роботи з даними.

ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	5
ВСТУП.....	6
1 ДОСЛІДЖЕННЯ СИСТЕМ УПРАВЛІННЯ МЕТАДАНИМИ.....	8
1.1 Вступ до предметної області	8
1.2 Актуальні проблеми	11
1.3 Методи опису метаданих.....	12
1.4 Огляд та аналіз систем	14
1.5 Висновки до розділу 1	21
2 ПОБУДОВА СИСТЕМИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ.....	24
2.1 Теоретична основа методу опису метаданих.....	24
2.2 Показники ефективності функціонування методу опису метаданих.....	27
2.3 Проектування макетів мобільного додатку	29
2.4 Архітектура системи	31
2.5 Опис архітектури сервера	34
2.6 Опис архітектури мобільного додатку	36
2.7 Опис інструментів розробки.....	37
2.8 Доступ до метаданих	39
2.9 Висновки до розділу 2.....	40
3 РОЗРОБКА МЕТОДУ ТА ОПТИМІЗАЦІЯ ЙОГО ЕФЕКТИВНОСТІ .	42
3.1 Метод опису метаданих для платформи BAF	42
3.2 Оптимізація використання HTTP сервісів та протоколу OData	56
3.3 Висновок до розділу 3	64
4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ТА ТЕСТУВАННЯ СИСТЕМИ.....	66

4.1	Тестування запуск серверної частини	66
4.2	Тестування та налаштування платформи VAF	68
4.3	Сценарії роботи користувача з системою	72
4.4	Оцінка швидкодії запропонованого методу	79
4.5	Висновок до розділу 4	84
ВИСНОВКИ		85
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ		87
ДОДАТОК А Лістинг програмного забезпечення мобільного додатку		95
ДОДАТОК Б Макети мобільного додатку		97
ДОДАТОК В Зображення у мобільному додатку		100
ДОДАТОК Г Тези		101
ДОДАТОК Г Презентація		103

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

CRM - Customer Relationship Management

BAF - Business Application Framework

ІС - Інформаційна система

ІТ - Інформаційні технології

SQL - Structured query language

ПЗ - програмне забезпечення

ІДЕ - Інтегроване середовище розробки

RTP - Realtime Transfer Protocol

RLS - Row Level Security

ВСТУП

У сучасному світі, де технології розвиваються зі швидкістю світла і конкуренція є нещадною, застосування мобільних додатків з інтегрованими системами управління набуває все більшої актуальності. Водночас, швидкий розвиток мобільних технологій і все більша потреба в доступності інформації "тут і зараз" зумовлюють актуальність розвитку мобільних додатків з CRM системами, які забезпечують нові можливості для залучення та утримання клієнтів. Ці інноваційні рішення відкривають нові можливості для покращення взаємодії з клієнтами, підвищення їхньої лояльності та оптимізації бізнес-процесів у різних галузях.

Бізнес-середовище сьогодні характеризується високою конкуренцією та швидкими змінами. Компанії прагнуть знайти ефективні способи залучення та утримання клієнтів, забезпечуючи високий рівень обслуговування та персоналізований підхід. Саме в цьому контексті мобільні додатки з CRM-системами стають невід'ємною частиною сучасних бізнес-стратегій, допомагаючи компаніям залишатися конкурентоспроможними та успішними.

Впровадження CRM-систем у мобільні додатки дозволяє співробітникам мати постійний доступ до актуальної інформації про клієнтів, незалежно від їхнього місцезнаходження. Це особливо важливо для компаній, чия діяльність пов'язана з частими виїздами або роботою в польових умовах, таких як сфера продажів, технічного обслуговування або консалтингу. Мобільні CRM-додатки забезпечують безперебійний обмін даними між співробітниками та центральною системою, дозволяючи оперативно відстежувати історію взаємодії з клієнтами, управляти замовленнями, планувати зустрічі та синхронізувати всі відповідні дані.

Однак, у міру того, як мобільні технології стають все більш поширеними, з'являється нагальна потреба в інтеграції CRM-систем з мобільними додатками. Мобільні додатки з CRM-функціоналом забезпечують безпрецедентну гнучкість та доступність, дозволяючи працівникам компанії отримувати доступ до важливих даних про клієнтів та управляти взаєминами з ними в будь-якому місці та в будь-

який час. Крім того, вони відкривають нові можливості для покращення клієнтського досвіду. Наприклад, у сфері роздрібно́ї торгівлі вони можуть використовуватися для надання персоналізованих рекомендацій та спеціальних пропозицій на основі історії покупок та переваг клієнтів. У сфері послуг ці додатки можуть забезпечувати швидкий доступ до інформації про призначені зустрічі, надавати можливість онлайн-бронювання та відстежувати статус обслуговування в режимі реального часу.

Використання мобільних CRM-додатків також сприяє підвищенню продуктивності та ефективності роботи співробітників. Завдяки інтегрованим функціям, таким як планування маршрутів, відстеження активності та автоматизація звітності, співробітники можуть ефективніше управляти своїм часом та ресурсами, зосереджуючись на найважливіших завданнях та забезпечуючи високий рівень обслуговування клієнтів.

За темою дипломної роботи опубліковані тези у матеріалах наукової конференції «СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ - 2024» (18-19 квітня 2024 р.) [37].

1 ДОСЛІДЖЕННЯ СИСТЕМ УПРАВЛІННЯ МЕТАДАНИМИ

1.1 Вступ до предметної області

Метадані - це інформація про самі дані. Вони надають адміністративні деталі, які допомагають користувачам краще розуміти та інтерпретувати набори даних. Додавання контексту та значення до метаданих полегшує пошук, організацію та інтерпретацію даних. Метадані є важливими у контролі якості отримання даних за таких причин:

1. Покращення розуміння даних: Вони допомагають зрозуміти структуру, взаємозв'язки та характеристики наборів даних. Аналіз метаданих дає організаціям уявлення про їхні інформаційні активи, що полегшує прийняття рішень щодо якості даних.

2. Забезпечення управління даними: Ефективне управління метаданими має важливе значення для управління даними в цілому. Метадані допомагають визначити правила та політику використання даних, забезпечуючи дотримання нормативних вимог. Вони також сприяють встановленню права власності та підзвітності за набори даних.

3. Сприяння інтеграції даних: Метадані відіграють ключову роль в узгодженні різних схем і визначень при інтеграції наборів даних з різних джерел. Вони полегшують інтеграцію, надаючи розуміння залучених елементів даних.

4. Виявлення даних: Метадані мають важливу роль у забезпеченні результативного пошуку, визначають конкретні атрибути полів, такі як назви наборів з даними, відомості про їх автора, часові мітки створення та призначені ключові теги. Це дозволяє користувачам швидко знаходити потрібну інформацію в великих колекціях даних.

Для опрацювання метаданих можна сформулювати відповідь на запит допомогою вбудованої мови програмування в платформі VAF. При цьому є можливість зручно отримувати доступ до тіла, заголовків та рядка вихідного запиту, а також можна формувати код, тіло та заголовки відповіді на свій розсуд [5].

Порівняно з наявними на платформі SOAP-веб-сервісами, HTTP-сервіси мають ряд переваг:

- простота розробки клієнтської частини;
- потенційно менший обсяг переданих даних;
- потенційно менше обчислювальне навантаження;
- HTTP-сервіси орієнтовані на ресурси, тоді як SOAP-сервіси орієнтовані на дії.

Перші три фактори особливо важливі для мобільних додатків. HTTP-сервіси можна використовувати як "легкі" RPC-сервіси, що не вимагають складної підготовки XML-пакетів. Методи можуть бути ідентифіковані в URL, а параметри можуть передаватися в опціях запиту або в його тілі. У другому випадку такі сервіси вже наближаються до SOAP, програючи йому в чіткості специфікації, але виграючи в гнучкості.

За своєю конструктивною реалізацією HTTP-сервіси дуже нагадують веб-сервіси, наявні на платформі. Точно так само існує спеціальний об'єкт конфігурації HTTP-сервісу.

Xamarin - це відносно нова технологія, заснована на принципах Microsoft. Проте на сьогодні її спільнота налічує вже понад 1,4 мільйона розробників.

Платформа Xamarin була створена розробниками відкритої платформи Mono, яка була реалізацією .NET і вперше представлена у 2001 році. На відміну від Mono, Xamarin спочатку існував виключно як комерційний проект, поки компанія не була придбана Microsoft.

Xamarin - це унікальний інструмент, оскільки для створення додатків на всіх мобільних платформах він використовує єдину мову - C#[21].

На відміну від рішень, нечутливих до мови розробки, таких як Appcelerator Titanium, платформа Xamarin дозволяє створювати нативні збірки для цільових платформ та розробляти високопродуктивні додатки з природним зовнішнім виглядом. З технічної точки зору, для кросплатформної розробки Xamarin використовує єдину мову C# та бібліотеки, обгорнуті в шар .NET. З точки зору

роботи та користувацького досвіду, такі додатки можна вважати нативними, тобто спеціально розробленими для iOS та Android[16].

Хоча код, пов'язаний з бізнес-логікою, доступом до бази даних та мережевими комунікаціями, може використовуватися на всіх платформах, Xamarin дозволяє створювати окремий шар коду інтерфейсу користувача для кожної конкретної платформи. Таким чином, кросплатформні додатки Xamarin виглядають на 100% нативними на будь-якому пристрої, забезпечуючи найкращий користувацький досвід порівняно зі стандартними гібридними додатками.

Платформа має дві основні форми: Xamarin.iOS та Xamarin.Android. У випадку з iOS, вихідний код компілюється безпосередньо в нативний асемблерний код ARM (AOT-компіляція), тоді як додатки Android Xamarin спочатку компілюються в проміжний універсальний код, а потім - в рідний під час виконання (JIT-компіляція). Однак цей підхід не є вирішенням усіх проблем. Як і будь-яка інша технологія, у Xamarin є свої переваги та недоліки, які останнім часом стали предметом гарячих дискусій.

Центральне місце в концепції захищених баз даних займає поняття інформаційної безпеки.

Інформаційна безпека - це захищеність інформації та інформаційної інфраструктури від випадкових або навмисних дій, природного чи штучного характеру, які можуть завдати неприйнятної шкоди суб'єктам інформаційних відносин, зокрема власникам і користувачам інформації та інфраструктури.

Це багатогранне поняття, вирішення якого потребує комплексного підходу. Ключовим принципом тут є "золоте правило забезпечення трьох китів" - доступності, цілісності та конфіденційності інформаційних ресурсів.

Доступність означає можливість отримати необхідну інформаційну послугу за прийнятний час. Цілісність передбачає актуальність, несуперечність інформації та її захищеність від руйнування і несанкціонованої зміни. Конфіденційність забезпечує захист від несанкціонованого доступу до інформації.

1.2 Актуальні проблеми

Розробка та впровадження мобільних додатків з інтегрованими системами управління відносинами з клієнтами (CRM) відкривають перед компаніями нові можливості для поліпшення ефективності бізнес-процесів, оптимізації взаємодії з клієнтами та збільшення лояльності. Однак, з цим також пов'язані певні виклики та актуальні проблеми, які потребують ретельного аналізу та вирішення [9]:

Забезпечення інформаційної безпеки та приватності даних: Із зростанням кількості зібраних даних зростає й ризик їх витоку або несанкціонованого доступу. Компаніям потрібно забезпечити високий рівень захисту персональних даних клієнтів, а також відповідати вимогам законодавства, особливо в контексті Генерального регламенту про захист даних (GDPR)[54].

Інтеграція з існуючими системами та інфраструктурою: Ефективна інтеграція мобільних додатків з CRM існуючими бізнес-процесами та системами, такими як ERP (Enterprise Resource Planning), складський облік, бухгалтерський облік тощо, є важливою умовою для досягнення максимальної користі та ефективності.

Управління змінами та адаптація персоналу: Впровадження нових технологій часто супроводжується необхідністю в адаптації та навчанні персоналу. Розвиток корпоративної культури, що сприяє інноваційності та гнучкості при користуванні новими інструментами, є ключовим для успішної інтеграції мобільних CRM.

Забезпечення високої продуктивності та доступності: Мобільні додатки повинні демонструвати стабільну роботу та високу швидкість реагування незалежно від кількості користувачів та обсягу даних. Це потребує використання сучасних технологій та вирішення проблем масштабування.

Створення зручного і інтуїтивно зрозумілого інтерфейсу: Дизайн мобільних додатків повинен бути не тільки привабливим, але й максимально зручним і зрозумілим для користувачів від різних цільових груп, що може стати непростим завданням [3].

Оцінка впливу на клієнтський досвід і поведінку: Важливо ретельно вимірювати та аналізувати, як впровадження мобільних CRM систем впливає на задоволеність клієнтів, лояльність і в кінцевому підсумку на доходи компанії.

Вирішення цих та інших актуальних проблем пов'язане з постійним вдосконаленням мобільних CRM систем, інвестуванням у новітні технології та методології, а також з розвитком гнучкості і інноваційності в бізнес-процесах. Тільки так компанії зможуть повною мірою скористатися перевагами мобільних CRM систем для підвищення своєї конкурентоспроможності у цифрову епоху.

1.3 Методи опису метаданих

Схема Дублінського ядра - це невеликий набір словникових термінів, які можна використовувати для опису різноманітних ресурсів.

Метадані Dublin Core можна застосовувати для багатьох цілей - від простого опису ресурсу до об'єднання словників метаданих різних стандартів та забезпечення сумісності для метаданих у зв'язаних даних і реалізаціях семантичної мережі[44].

Термін "метадані" означає "дані про дані". Метадані описують контекст об'єктів, таких як файли документів, зображення, аудіо- та відеоматеріали. Ця традиція опису ресурсів бере початок з найдавніших архівів і бібліотечних каталогів. Сучасне поле "метаданих", що дало початок Dublin Core та іншим нещодавнім стандартам, з'явилося разом із веб-революцією середини 1990-х років.

"Дублін" відноситься до Дубліна, штат Огайо, США, де схема виникла під час запрошення OCLC/NCSA Metadata Workshop 1995 року, організованого Центром комп'ютерних бібліотек онлайн (OCLC) та Національним центром суперкомп'ютерних застосувань (NCSA). "Ядро" означає, що терміни метаданих є широкими та загальними, придатними для опису широкого діапазону ресурсів. Семантика Dublin Core була розроблена та підтримується міжнародною міждисциплінарною групою професіоналів із бібліотечної справи, інформатики, кодування тексту, музеїв та інших суміжних галузей.

Dublin Core Metadata Initiative (DCMI) забезпечує відкритий доступ для розробки сумісних стандартів онлайн-метаданих для широкого спектру цілей і бізнес-моделей. Діяльність DCMI включає консенсусні робочі групи, глобальні конференції та семінари, зв'язок зі стандартами та освітні заходи для сприяння широкому прийняттю стандартів і практики метаданих.

MARC XML. Управління розвитку мережі та стандартів MARC розробило структуру [49] для роботи з даними MARC у середовищі XML. Ця структура має бути гнучкою та розширюваною, щоб дозволити користувачам працювати з даними MARC відповідно до їхніх потреб. Структура (рисунок 1.1) містить багато компонентів, таких як схеми, таблиці. Також доступні утиліти для перетворення між MARC і MARC XML.

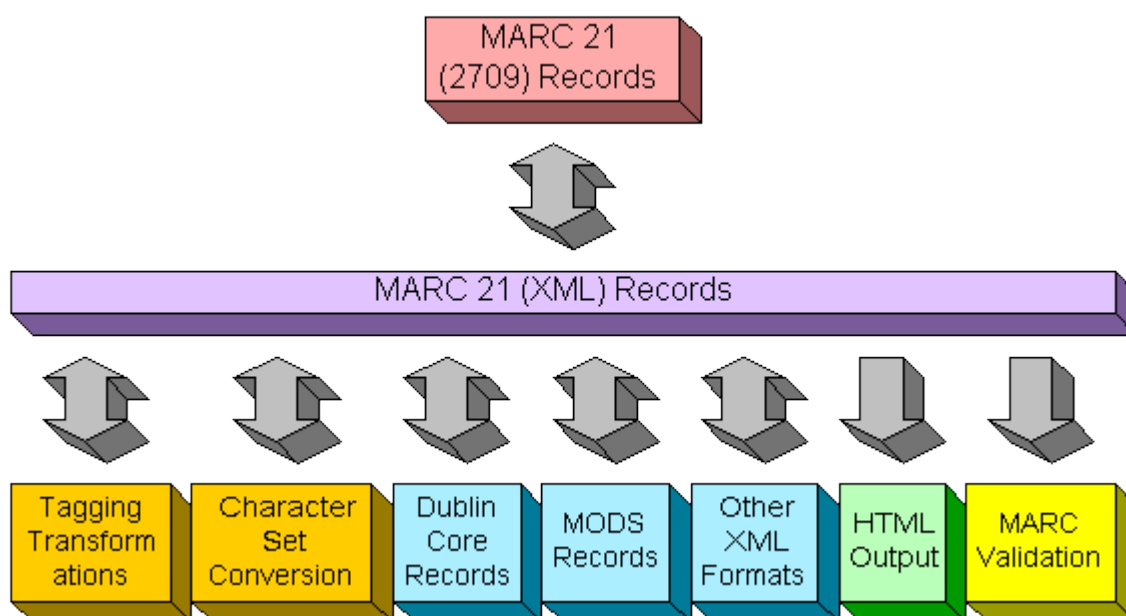


Рисунок 1.1 – . Структура MARC XML

Метадані доступності допомагають користувачам знаходити ресурси, які відповідають їхнім конкретним потребам у доступності. Ці метадані повідомляють користувачу, які функції доступності має ресурс, наприклад, чи можна змінювати розмір тексту під час читання електронної книги, чи ресурс має миготливі небезпеки, які можуть вплинути на деяких відвідувачів.

Для обміну інформацією про доступність між видавцями, роздрібними торговцями та дистриб'юторами книжкова індустрія використовує формат даних ONIX (ONIX for Book Product Information). ONIX використовує контрольовані словники, відомі як списки кодів. Список кодів 196 призначений для деталей доступності електронної публікації. ONIX передається як XML-файл всередині книги або разом з нею.

Schema.org також створює стандартну мову для веб-вмісту, включаючи схему для метаданих доступності. Видавці надсилають метадані доступності ONIX і schema.org разом зі своїми цифровими публікаціями. Однак постачальники та розповсюджувачі не завжди включають ці метадані у записи MARC, які вони надають бібліотекам. Коли це станеться, можна переглянути файл .opf EPUB, щоб отримати метадані schema.org і використати перехід до формату MARC.

Оскільки метадані доступності вже створені видавцем, немає необхідності відкривати файл .opf і додавати їх до запису MARC. Щоб надати доступ до інформації про доступність, яку надають видавці, бібліотеки повинні просити постачальників/розповсюджувачів включати, відображати та забезпечувати можливість пошуку/фільтрування всіх метаданих доступності.

Практика включення метаданих доступності в бібліографічні записи MARC розвивається. У 2018 році MARC21 додав два поля: 341 Accessibility Content і 532 Accessibility Note. Бібліотеки повинні використовувати ці теги, коли це можливо, хоча наразі поле 341 може бути недоступним для пошуку в деяких інтегрованих бібліотечних системах. Бібліотеки повинні попросити OCLC та постачальників ILS зробити ці поля доступними для пошуку.

1.4 Огляд та аналіз систем

KeyCRM є українською CRM-системою, призначеною для бізнесів різних масштабів, включаючи малий, середній та великий бізнес. Ця система охоплює широкий спектр функціональності, включаючи управління лідами та заявками,

спілкування з клієнтами, управління замовленнями та покупцями, а також надає інструменти для аналітики та звітності[18].

Основні можливості KeyCRM включають:

- організація роботи з клієнтами: збір та обробка лідів, замовлень, записів на послугу з месенджерів, сайтів, соціальних мереж;
- інтеграція з платформами та сервісами: широке покриття інтеграції з маркетплейсами, соціальними мережами, месенджерами, службами доставки та платіжними системами;
- функціонал для торгівлі: управління складом, модуль для підключення постачальників, воронки продажів і лідів;
- посилення безпеки і контролю доступу: можливість детально налаштовувати права доступу співробітників;
- мобільний додаток: дозволяє управляти бізнесом зі смартфона, незважаючи на те, що деякі розділи все ще знаходяться в розробці.

Користувачам KeyCRM пропонуються такі можливості:

- керування замовленнями з усіх маркетплейсів у єдиному форматі;
- інтеграція з Instagram для керування повідомленнями, коментарями та воронками продажів;
- підключення сайтів через різні CMS;
- інструменти для збору заявок, управління проектами, воронками угод;
- удосконалене спілкування через вбудовані інструменти телефонії та месенджерів.

Переваги KeyCRM:

- гнучкість та адаптованість до потреб різних видів бізнесу;
- широкий спектр інтеграцій, які надають можливість автоматизації багатьох процесів;
- мобільність, дозволяючи керувати бізнесом з будь-якого місця;
- детальна аналітика та звітність допомагають оптимізувати воронки продажів і підвищити ефективність бізнес-процесів.

Скріншоти додатку подано на рисунку 1.2.

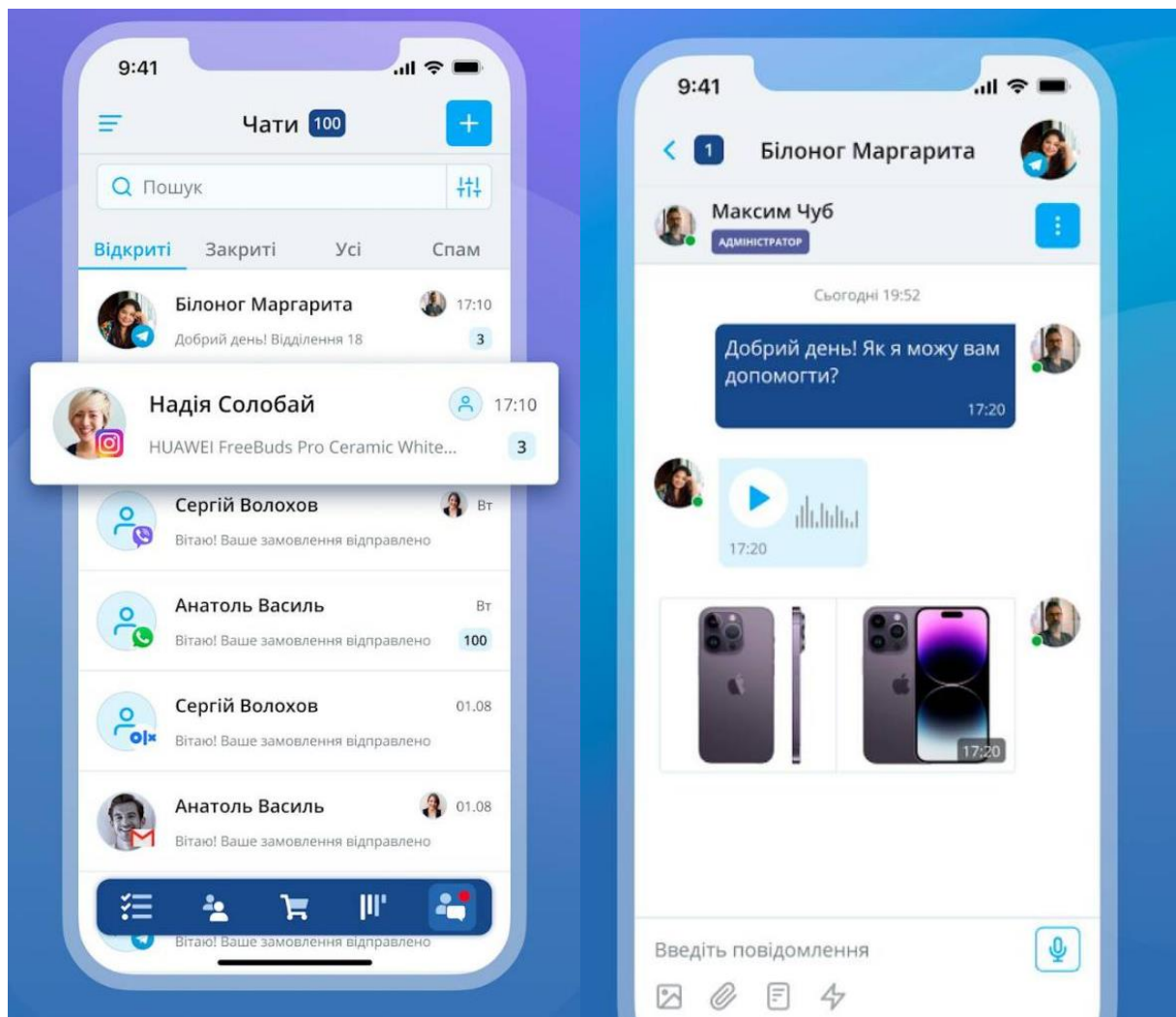


Рисунок 1.2 – Скріншоти додатку SAP CRM Service Manager

Недоліки KeyCRM:

- для малого та середнього бізнесу вартість може бути досить високою, особливо при необхідності великої кількості додаткових інтеграцій та підключення великої кількості користувачів;
- з огляду на широкий функціонал та гнучкість системи, процес налаштування KeyCRM під специфічні потреби компанії може виявитися досить складним, вимагаючи часу і, можливо, залучення ІТ-спеціалістів;
- для ефективного використання всіх можливостей CRM системи персонал потребує додаткового навчання, що також займає час і ресурси;

- оскільки вказано, що частина розділів мобільної версії знаходиться у розробці, це може обмежувати повноцінне використання системи "на ходу" до їх повного впровадження;

- широкі інтеграції також означають залежність від роботи зовнішніх сервісів та платформ, що може призводити до певних труднощів у випадку їх змін чи недоступності.

SAP CRM Service Manager - це потужне рішення для управління взаємовідносинами з клієнтами та обслуговуванням, що входить до складу SAP Customer Relationship Management (SAP CRM). Скріншоти додатку подано на рисунку 1.3.

Розглянемо основні можливості та особливості цієї системи:

- дозволяє ефективно обробляти запити клієнтів, створювати та відстежувати заявки на обслуговування, а також контролювати весь цикл вирішення проблем;

- система включає потужний контактний центр, який забезпечує централізоване управління всіма каналами взаємодії з клієнтами, такими як телефон, електронна пошта, веб-портал тощо;

- можливість створювати та відстежувати контракти на обслуговування, визначати та контролювати дотримання угод про рівень обслуговування (SLA);

- дозволяє ефективно планувати та розподіляти ресурси для виконання завдань з обслуговування, враховуючи навички, навантаження та розташування співробітників;

- включає базу знань, де зберігається інформація про продукти, рішення проблем та найкращі практики. Клієнти можуть використовувати портал самообслуговування для пошуку відповідей на свої запитання;

- тісна інтеграція з іншими модулями SAP, такими як SAP ERP, SAP Supply Chain Management, що забезпечує безперебійний обмін даними та уніфіковане середовище для управління бізнес-процесами.

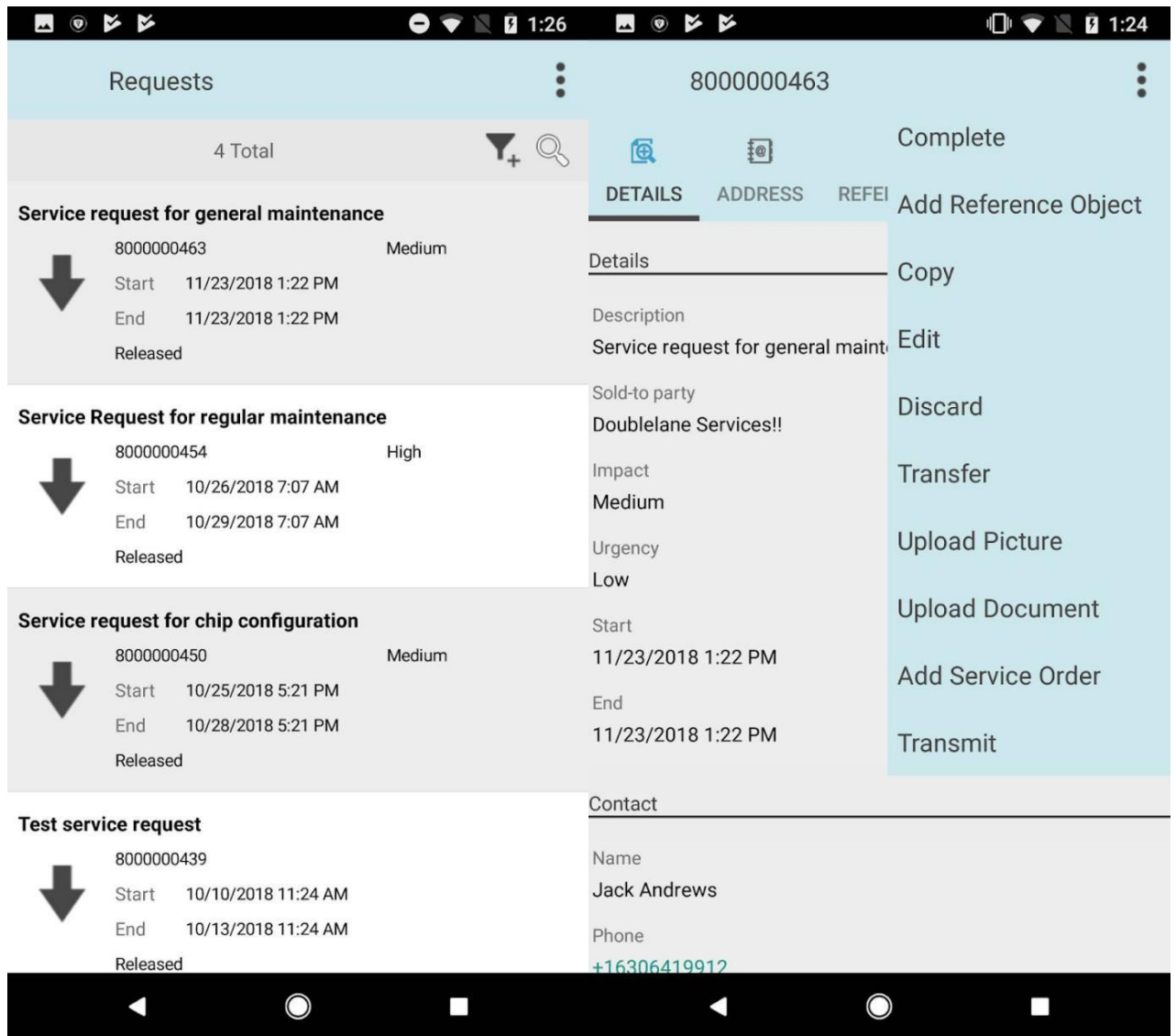


Рисунок 1.3 – Скріншоти додатку SAP CRM Service Manager

Переваги SAP CRM Service Manager:

- потужна функціональність для управління обслуговуванням клієнтів на всіх етапах;
- централізоване управління всіма каналами взаємодії з клієнтами;
- ефективне планування та розподіл ресурсів;
- аналітичні інструменти для відстеження та вдосконалення рівня обслуговування;
- інтеграція з іншими модулями SAP для уніфікованого управління бізнес-процесами.

Недоліки:

- висока вартість ліцензій та впровадження;
- складність налаштування та інтеграції з іншими системами;
- необхідність спеціалізованого навчання персоналу;
- обмежена гнучкість для швидкої адаптації до змін у бізнес-процесах.

Мобільна платформа VAF - це програмний продукт від компанії "VAF", який дозволяє створювати мобільні додатки для автоматизації бізнес-процесів, зокрема для використання як CRM. Розглянемо її основні можливості та переваги в якості CRM-системи[14].

Додаток можна встановити на смартфони та планшети з операційними системами iOS та Android, що дозволяє працівникам компанії працювати з даними в будь-якому місці та в будь-який час.

Платформа повністю інтегрована з типовими конфігураціями VAF, такими як "Управління торгівлею", "ERP", "УНФ" тощо. Це забезпечує безперебійний обмін даними між мобільним додатком та основною інформаційною базою VAF.

Мобільний додаток дозволяє зберігати та редагувати інформацію про клієнтів, контакти, історію взаємодії, угоди та замовлення прямо з мобільного пристрою.

Працівники можуть планувати зустрічі, візити до клієнтів, ставити завдання та контролювати їх виконання через мобільний додаток.

Система надає аналітичні інструменти для відстеження продажів, ефективності роботи співробітників, формування різноманітних звітів та графіків.

Платформа дозволяє налаштовувати мобільний додаток відповідно до специфіки бізнесу та потреб компанії. Дані можна редагувати та синхронізувати в офлайн-режимі, коли немає доступу до інтернету.

Основними перевагами "VAF мобільної платформи" як CRM-системи є мобільність, інтеграція з VAF, повнофункціональне управління клієнтською базою та можливість налаштування під конкретні бізнес-процеси компанії. Проте слід зазначити, що вона може бути не такою гнучкою та масштабованою, як спеціалізовані CRM-системи від провідних постачальників. Скріншоти додатку подано на рисунку 1.4.

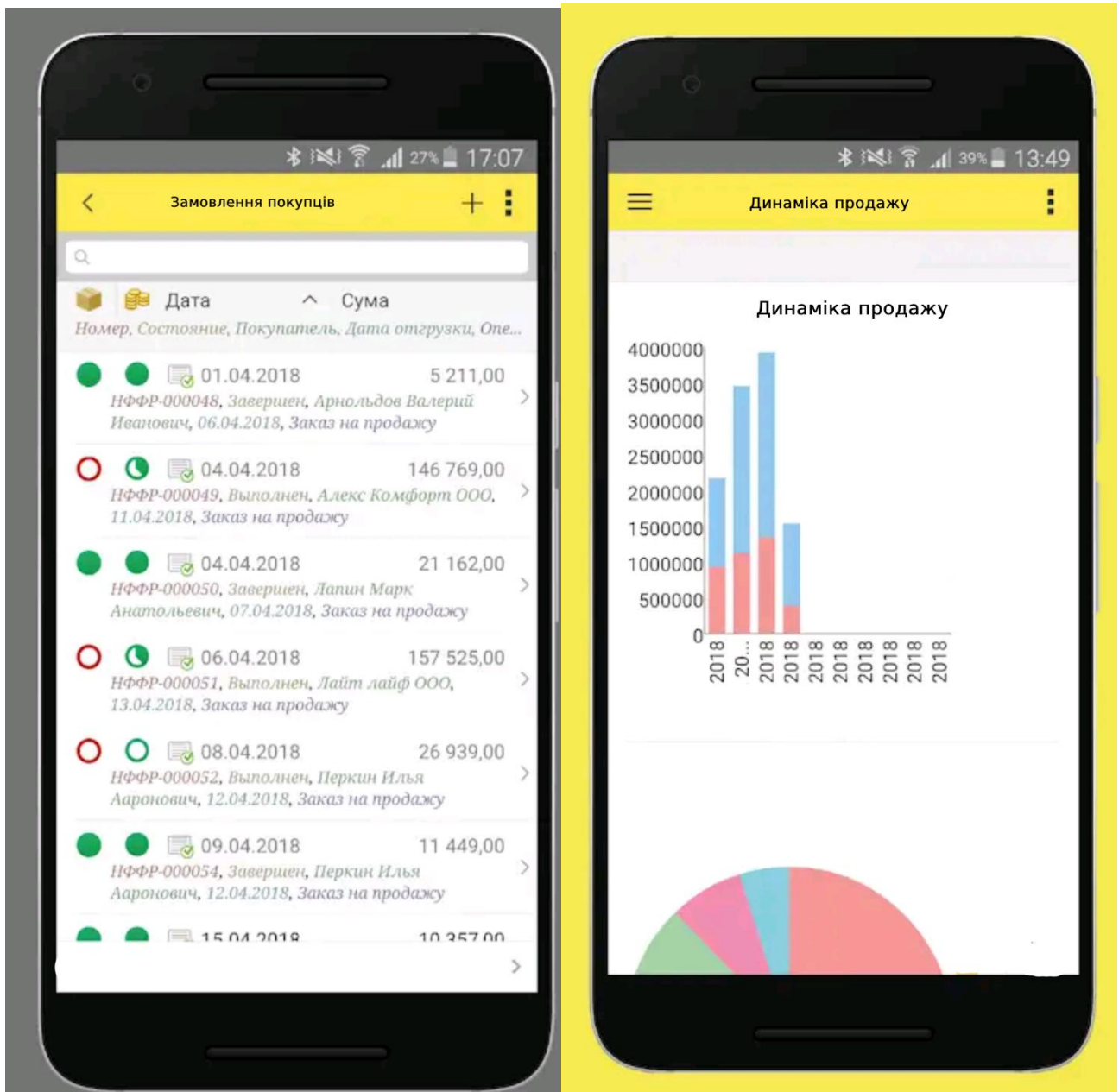


Рисунок 1.4 – Скріншоти мобільної платформи ВАФ

Переваги "VAF мобільної платформи" як CRM-системи:

- тісна інтеграція з типовими конфігураціями ВАФ, такими як "Управління торгівлею", "Бухгалтерія" тощо. Це забезпечує безшовний обмін даними між мобільним додатком та основною інформаційною базою ВАФ, уникаючи дублювання даних та необхідності синхронізації з іншими системами;
- можливість налаштувати мобільний додаток відповідно до специфіки бізнесу та потреб компанії, адаптувати його під існуючі бізнес-процеси та вимоги;

- працівники можуть працювати з даними в режимі офлайн, коли немає доступу до інтернету, а потім синхронізувати зміни при з'єднанні;
- функції для планування зустрічей, візитів до клієнтів, постановки завдань та контролю їх виконання прямо з мобільного додатку;
- для компаній, які вже використовують продукти VAF, вартість впровадження мобільної платформи може бути нижчою, ніж придбання окремої CRM-системи.

Недоліки "VAF мобільної платформи" як CRM-системи:

- у порівнянні зі спеціалізованими CRM-системами, "VAF мобільна платформа" може мати обмежену функціональність та менше можливостей для управління взаємовідносинами з клієнтами;
- для великих компаній з великою кількістю користувачів та складними бізнес-процесами, мобільна платформа може виявитися недостатньо масштабованою та гнучкою.

Оскільки система тісно інтегрована з продуктами VAF, компанії, які не використовують ці продукти, можуть зіткнутися з труднощами при впровадженні та інтеграції. Платформа дозволяє певну кастомізацію, вона може бути обмеженою у порівнянні з можливостями налаштування спеціалізованих CRM-систем.

1.5 Висновки до розділу 1

Розглядаючи становище та проблематику мобільних кросплатформних CRM-систем, особливо зосереджуючись на методах опису метаданих об'єктів на платформі VAF, ми зіткнулися з комплексом викликів та перспектив, які передбачають значний імпакт на сучасний бізнес та взаємодію з клієнтами. Інноваційні CRM-системи, що підтримують мобільні технології та кросплатформність, відкривають перед компаніями нові горизонти для підвищення ефективності, лояльності клієнтів та загальної конкурентоспроможності. Незважаючи на технічні й організаційні складнощі, включно з питаннями забезпечення безпеки даних, інтеграції з наявними системами, управління змінами,

забезпеченням високої продуктивності та створенням інтуїтивно зрозумілого користувацького інтерфейсу, впровадження таких систем є не лише бажаним, але й необхідним кроком для прогресивних компаній.

Розробка ефективної CRM-системи є складним і багатограним завданням, яке вимагає ретельного планування, глибокого розуміння потреб клієнтів та використання передових технологій. Проаналізувавши такі рішення, як KeyCRM, "BAF мобільна платформа", SAP CRM Service Manager, можна виділити кілька ключових аспектів, на які слід звернути увагу.

Гнучкість та масштабованість архітектури системи є критично важливими. CRM-рішення повинні легко адаптуватися до мінливих вимог бізнесу, забезпечувати безперебійну роботу при зростанні кількості користувачів та обсягів даних. Використання сучасних хмарних технологій, мікросервісної архітектури та контейнеризації може допомогти досягти цієї мети.

По-друге, інтеграція з іншими системами та платформами стає все більш важливою. Клієнти очікують безшовного обміну даними між CRM, ERP, електронною комерцією, соціальними мережами та іншими сервісами. Розробникам слід приділяти особливу увагу створенню відкритих API та використанню стандартів інтеграції.

По-третє, забезпечення високого рівня безпеки та конфіденційності даних є обов'язковою вимогою. Розробники повинні дотримуватися найсуворіших стандартів захисту інформації, шифрування даних, контролю доступу та відповідності нормативним вимогам.

Четвертим ключовим аспектом є зручність використання та інтуїтивний інтерфейс. CRM-система повинна бути простою у навчанні та використанні, забезпечувати ефективну навігацію та візуалізацію даних. Застосування принципів користувацького досвіду (UX) та залучення користувачів на етапі проектування є критично важливими.

Загалом, створення успішної CRM-системи вимагає від розробників поєднання технічної майстерності, розуміння бізнес-процесів та орієнтації на потреби клієнтів. Лише постійно вдосконалюючи свої рішення та слідкуючи за

новітніми тенденціями, розробники зможуть забезпечити своїм клієнтам конкурентні переваги в управлінні взаємовідносинами з клієнтами.

2 ПОБУДОВА СИСТЕМИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ

2.1 Теоретична основа методу опису метаданих

Для опису структури методу опису метаданих може бути використаний апарат алгебри каскадів. Модель предметної області повинна забезпечувати формалізоване представлення (логічне, алгебраїчне і т.п.) досліджуваних елементів та їх взаємозв'язків. Припускаємо, що об'єкти утворюють множину $X = \{X_j\}$, на якій задано взаємозв'язки $R = \{R_i\}$, де з кожним взаємозв'язком створюється поняття арності – $\text{arn}(R_i)$, тобто кількість об'єктів, які приймають участь у факті, описуваному за допомогою цього взаємозв'язку.

На основі множини X і додатково введеної множини $\Theta = \{Q_i\}$, елементи якої назвемо зв'язками, будується множина Y (назвемо її носієм множини каскадів) за наступними правилами:

- якщо $y_i \in X$, то $y_i \in Y$;
- якщо $y_1, y_2 \in Y, Q_j \in \Theta$ то $y_1 Q_{jy_2} \in Y$;
- $\Lambda \in Y$ (Λ – пусте ім'я).

Описані вище правила призводять до алгебри слів, або термів, однак не в загальній постановці, а в спеціальній формі, зручній для побудови апарату каскадів.

Каскадом називатимемо елемент множини:

$$K = \bigcup_{m=0}^{n-1} Y^m \times B(R \times B(Y^{n-m})),$$

де Y^k – k -кратний декартовий добуток,

$B(A)$ – булеан для множини A .

Легко помітити, що K має із множини:

$$K^m = Y^m \times B(R \times B(R \times B(Y^{n-m})),$$

які будемо називати m -основними системами каскадів. Число утворюючих m -основних систем залежить від максимальної арності взаємозв'язків, що входять в множину R .

Будемо використовувати наступне відображення каскаду:

$$\begin{cases} K_1 = \{y_j^m, \{R_1^i, A_{ij1}\}, \dots, \{R_s^i A_{ijs1}\}\} \\ A_{ijp} = \{y_{ijpk}^{n-m}\}, k = 1, \dots, t_{ijp} \\ y_{ijpk}^{n-m} = \{y_{m+1}^{ijpk}, \dots, y_n^{ijpk}\}, \end{cases}$$

При цьому $R_j \in R(K_1)$, $R(K_1)$ – множина взаємозв'язків каскаду K_1 .

Визначено морфізм з K^m в K^{m-1} , властивості якого роблять достатнім розгляд властивостей певні m -основної системи, які за розглянутим аморфізмом розповсюджуються на інші системи множини каскадів. Для визначення операцій алгебри каскадів використаємо множину зв'язок $(.,!-)$.

При цьому вводяться наступні еквівалентні перетворення складних імен:

- 1) $y_i y_j = y_j y_i$;
- 2) $y_i (y_j y_k) = (y_i y_j) y_k$;
- 3) $y_i \Lambda = y_i$;
- 4) $y_i ! y_j = y_j ! y_i$;
- 5) $y_i ! (y_j ! y_k) = (y_i ! y_j) ! y_k$;
- 6) $y_i ! \Lambda = \Lambda$;
- 7) $y_i - \Lambda = y_i$;
- 8) $y_i - (y_j ! y_k) = (y_i - y_j) (y_i - y_k)$;
- 9) $y_i - (y_j y_k) = (y_i - y_j) ! (y_i - y_k)$;
- 10) $y_i y_i = y_i$;
- 11) $y_i ! y_i = y_i$;
- 12) $y_i - y_i = y_k$;

В результаті множини каскадів з операціями об'єднання, перетину і доповнення і каскадами одиничним (утворюється як об'єднання всіх одиничних каскадів) і універсальним формують булеву алгебру. Описані вище операції

відображають класичний підхід до створення алгебраїчних систем. Введення ж операцій, які розглядаються далі, викликано саме характерними особливостями предметної області проектування АСУ гнучкими продуктивними системами.

При використанні апарату числення предикатів першого порядку для реалізації пошуку, наприклад, всіх структурних елементів, які забезпечують дані, нам довелося б багаторазово будувати дедуктивний висновок, вирішуючи паралельно предикатні рівняння. В рамках теорії каскадів це можна здійснити введенням спеціальних операцій, і в першу чергу операцій приєднання і транзитивного замикання.

ЯКЩО <УМОВА> ТО <ДІЯ>

Всі правила, які реалізуються в рамках даної теорії, можуть бути розбиті на дві підгрупи. Одну утворюють правила, в <УМОВА> і <ДІЯ> яких входять лише явні посилання на існуючу реалізацію каскадної бази знань. Їх назвемо явними, або нуль-правилами.

Інша група представлена так званими параметричними правилами, тобто <УМОВА> та/або <ДІЯ> містять неявні величини (з точки зору лямбда-числення - вільні змінні). Явні правила є однією з головних характеристик конкретної предметної області, водночас як частина параметричних правил кожне з яких може бути спільним для багатьох предметних областей. Основу, що включається в правило продукції, становить предикат існування каскаду : утворюються у вигляді функціональних виразів над каскадною базою знань, при цьому використовуються описані вище операції над каскадами.

Основу виведення утворює банк стратегій, який включає в себе існуючі моделі виведення: метод резолюцій, дедукцію та ін., в тому числі моделі виведення, характерні для процедур проектування.

Основними операціями, визначеними на множині стратегій, є:

- запуск дії;
- визначання дії;
- умова-перехід;
- визначення стратегії;

- завершення виведення.

2.2 Показники ефективності функціонування методу опису метаданих

Ефективне управління метаданими є ключовим аспектом успішного функціонування сучасних інформаційних систем. Для забезпечення високої якості та оптимального використання метаданих необхідно розробити комплексну систему оцінювання їх ефективності. У рамках магістерської роботи, присвяченої вдосконаленню методу опису метаданих, пропонується використання низки математичних показників, що дозволять всебічно проаналізувати різні аспекти ефективності цього процесу.

Розроблені формули охоплюють такі ключові характеристики метаданих, як повнота, точність, актуальність, доступність, сумісність та ефективність витрат. Застосування цих показників дає можливість кількісно оцінити, наскільки повно та коректно описані метадані, наскільки вони відповідають поточному стану інформаційних ресурсів, наскільки легко користувачі можуть їх знайти та використати, а також визначити економічну доцільність витрат на підтримку системи метаданих.

Для оцінки ефективності функціонування методу опису метаданих потрібно використовувати такі показники:

1. Коефіцієнт повноти метаданих характеризує частку описаних елементів метаданих від загальної кількості елементів, що потребують опису (КПМ):

$$\text{КПМ} = N^{\text{ОМ}} / N^{\text{ЗМ}}$$

де $N^{\text{ОМ}}$ - кількість описаних елементів метаданих;

$N^{\text{ЗМ}}$ - загальна кількість елементів метаданих, що потребують опису.

2. Коефіцієнт точності метаданих показує частку вірно описаних елементів метаданих від загальної кількості описаних (КТМ):

$$\text{КПМ} = N^{\text{ВОМ}} / N^{\text{ЗМ}}$$

де $N^{\text{ВОМ}}$ - кількість вірно описаних елементів метаданих;

$N^{\text{ЗМ}}$ - загальна кількість описаних елементів метаданих.

3. Коефіцієнт актуальності метаданих визначає частку актуальних елементів метаданих від загальної кількості описаних (КАМ):

$$\text{КАМ} = N^{\text{АМ}} / N^{\text{ОМ}}$$

де $N^{\text{АМ}}$ - кількість актуальних елементів метаданих;

$N^{\text{ОМ}}$ - загальна кількість описаних елементів метаданих.

4. Коефіцієнт доступності метаданих характеризує частку користувачів, які змогли отримати необхідні метадані, від загальної кількості користувачів, що їх потребували (КДМ):

$$\text{КДМ} = N^{\text{ОМ}} / N^{\text{ЗК}}$$

де $N^{\text{ОМ}}$ - кількість користувачів, які змогли знайти та отримати необхідні метадані;

$N^{\text{ЗК}}$ - загальна кількість користувачів, що потребували метадані.

5. Коефіцієнт сумісності метаданих показує частку метаданих, що відповідають стандартам та можуть бути інтегровані, від загальної кількості описаних (КСМ):

$$\text{КСМ} = N^{\text{СМ}} / N^{\text{ОМ}}$$

де $N^{\text{СМ}}$ - кількість метаданих, що відповідають стандартам та можуть бути інтегровані;

$N^{\text{ОМ}}$ - загальна кількість описаних елементів метаданих.

6. Коефіцієнт ефективності витрат на метадані характеризує співвідношення отриманих вигід до понесених витрат на метадані (КЕВМ):

$$\text{КЕВМ} = N^B / N^3$$

де N^B - отримані вигоди від використання метаданих;

N^3 - витрати на створення та підтримку метаданих.

Комплексний аналіз запропонованих коефіцієнтів дозволить виявити сильні та слабкі сторони існуючого методу опису метаданих, окреслити перспективні напрями його вдосконалення та розробити обґрунтовані рекомендації щодо підвищення ефективності управління метаданими в рамках цієї роботи. Застосування цих математичних інструментів сприятиме підвищенню якості, доступності та цінності метаданих як важливого інформаційного ресурсу.

2.3 Проектування макетів мобільного додатку

Для ефективного проектування інтерфейсу мобільного додатку важливо використовувати спеціалізоване програмне забезпечення, яке дозволяє створювати високоякісні та інтуїтивно зрозумілі макети. Одним з найпопулярніших інструментів для цього є Figma - хмарний графічний редактор, що надає широкі можливості для дизайну, прототипування та тестування інтерфейсів[4].

При проектуванні макетів мобільного додатку в Figma важливо дотримуватися таких ключових принципів:

1. Адаптивність: Макети, який будується повинен коректно відображатися на різних розмірах екранів мобільних пристроїв. Використовуйте сітки, компоненти та інші інструменти Figma для забезпечення адаптивності.

2. Юзабіліті: Варто приділити увагу зручності використання додатку. Розміщувати основні елементи управління в легкодоступних місцях, дотримуйтесь принципів ергономіки та інтуїтивної навігації.

3. Візуальна привабливість: Створювати естетично приємні та гармонійні макети, використовуючи продуману колірну палітру, типографіку та графічні елементи.

4. Прототипування: Створювати інтерактивні прототипи для демонстрації функціональності додатку та тестування його юзабіліті.

5. Сумісна робота: Можливості Figma для спільної роботи над проектом, обговорення ідей та отримання зворотного зв'язку від команди.

Макети відображають:

- розподіл просторів екрану;
- пріоритетний контент;
- функціонал додатку;
- можливі дії користувача;
- відношення між екранами.

Макети не відображають:

- елементи дизайну;
- кольори;
- фактичні зображення;
- шрифти;
- логотипи.

Виділимо основні переваги розробки та використання макетів додатка:

- надають чітке уявлення про те, як буде виглядати додаток та як він буде працювати;
- дозволяють швидко змінювати структуру проекту, не треба перебудовувати код проекту;
- зменшують загальну вартість проекту.
- зібравши продуманий макет можна оцінити й виправити недоліки на ранньому етапі проектування, що буде коштувати значно дешевше, ніж зміни в кінцевому продукті;
- надають розуміння обсягів запланованих робіт. Макет можна розділити за логічними частинами й оцінити складність кожної окремо;

- макет може виступати в ролі прототипу майбутнього продукту, який можна використати у презентаціях для кінцевого користувача або замовника.

Застосування Figma (рисунок 2.1) дозволяє значно підвищити ефективність процесу проектування мобільних додатків, забезпечити високу якість макетів та прискорити розробку готового продукту. Розроблені макети Figma, подано у додатку А.

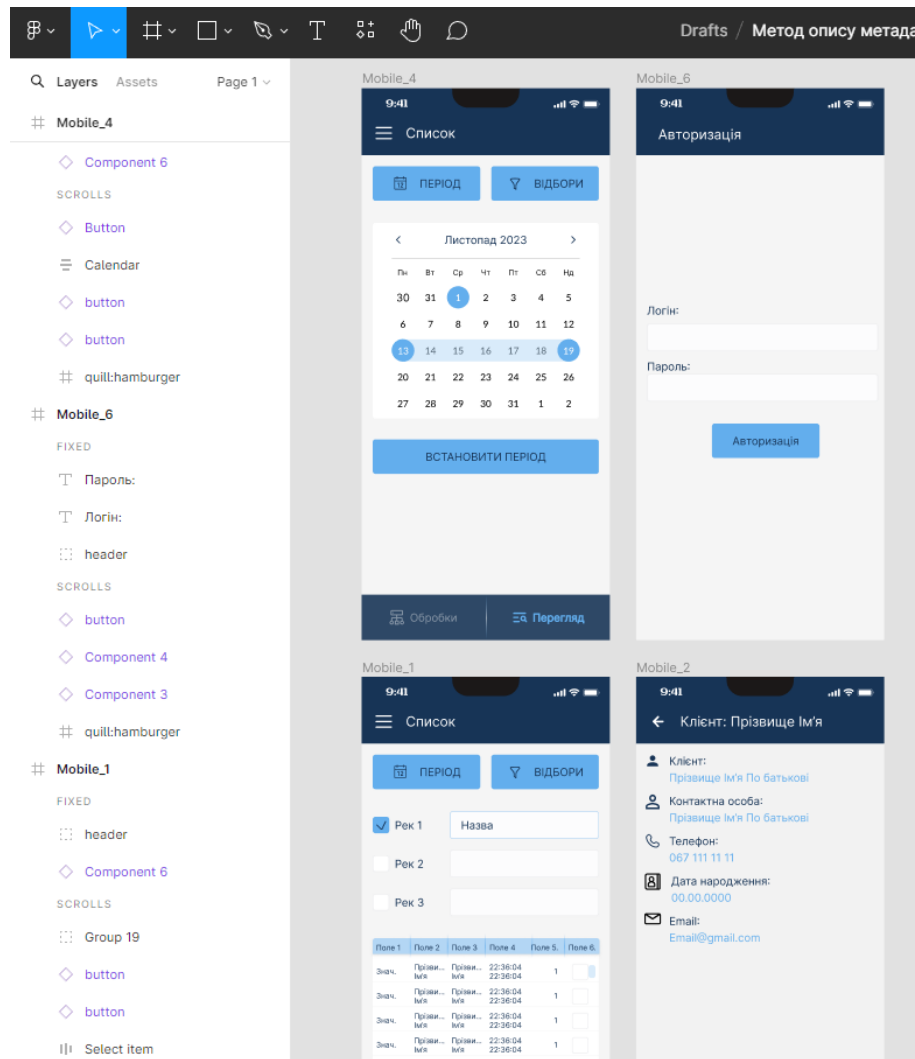


Рисунок 2.1 – Побудова макету в Figma

2.4 Архітектура системи

Елемент системи база даних Microsoft SQL Server. Вона дозволяє ефективно зберігати та обробляти великі обсяги даних. Це особливо важливо для великих підприємств, які працюють з великими обсягами інформації. Її потужний механізм

оптимізації запитів, індексації та кешування даних, що забезпечує високу продуктивність при виконанні запитів та операцій обробки даних[19].

SQL Server має широкий спектр механізмів для забезпечення безпеки даних, включаючи аутентифікацію, авторизацію, шифрування та механізми резервного копіювання даних, що робить його надійним вибором для зберігання важливої інформації. Завдяки різноманітним можливостям інтеграції, таким як підтримка стандартних протоколів зв'язку та інструменти для роботи з зовнішніми даними, SQL Server забезпечує сумісність з іншими системами, що спрощує інтеграцію з існуючими рішеннями для платформи BAF. База даних використовуються для зберігання та керування даними, що використовуються в CRM системі та мобільному додатку. Вона забезпечують надійне та структуроване зберігання інформації.

Елемент системи CRM система на платформі BAF. Центральним модулем системи є CRM система на платформі BAF вона це основним компонентом системи, де здійснюється обробка даних про клієнтів, продажі, контакти в залежності від використовуваної конфігурації тощо. Дає можливість налаштовувати інтерфейс системи, списків та елементів. Цей елемент гарантує, що інформація завжди зберігається в актуальному стані.

Вона використовує HTTP сервіси для обміну даними з мобільним додатком, забезпечуючи безпеку та швидкість обміну інформацією.

Елемент системи HTTP Сервіси. HTTP сервіси представляють собою серверні компоненти, які надають можливість мобільним додаткам взаємодіяти з CRM системою через мережу Інтернет. Вони сервіси гарантують безпечний та швидкий обмін інформацією між мобільним додатком та CRM системою, використовуючи сучасні методи шифрування та безпеки. HTTP-сервіси для BAF підтримують різні типи REST запитів, такі як GET, POST, PUT та DELETE, що дозволяє виконувати різноманітні операції з даними інформаційної бази BAF і бути представленими у стандартних форматах обміну даними, таких як JSON або XML, що дозволяє легко інтегруватися з будь якою системою.

Елемент системи Мобільний додаток (Xamarin). Інтерфейс користувача мобільного додаток розроблятиметься з використанням мови програмування Xamarin і матиме інтуїтивний та зручний інтерфейс для користувачів. Користувачі можуть легко навігувати, використовуючи панелі, кнопки та меню.

Додаток надає можливість користувачам вводити, редагувати та переглядати дані про клієнтів, продажі, контакти та іншу інформацію.

Користувачі можуть швидко здійснювати операції з даними, не використовуючи ПК версію.

Він використовує HTTP сервіси для безпечного обміну даними з CRM системою через Інтернет, що дозволяє користувачам завжди мати доступ до актуальної інформації.

Ці компоненти працюють разом для забезпечення безперебійної та ефективної роботи системи, що дозволяє підтримувати високий рівень обслуговування системи та оптимізувати робочі процеси (рисунок 2.2).

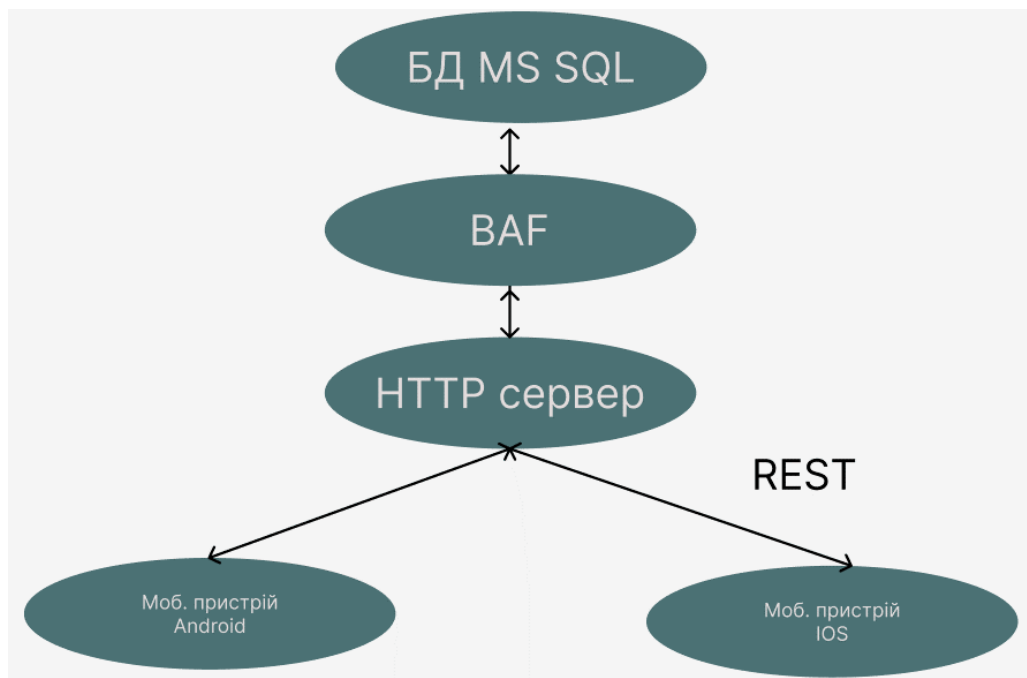


Рисунок 2.2 – Схема клієнт серверної архітектури програмного комплексу

2.5 Опис архітектури сервера

При побудові сервера спочатку потрібно визначитися з його архітектурою, кількістю її шарів та відповідальністю кожного[24].

Сформуємо функціоналу, який повинен виконувати сервер:

- обробляти запити додатку та повертати відповіді у форматі json;
- механізми обробки помилок, що повертає повідомлення з кодом помилок клієнту;
- управління транзакціями буде здійснювати платформа BAF;
- треба обробляти авторизацію та автентифікацію користувача;
- містити налаштування мобільного додатку в платформі BAF;
- можливість підключення до зовнішніх ресурсів.

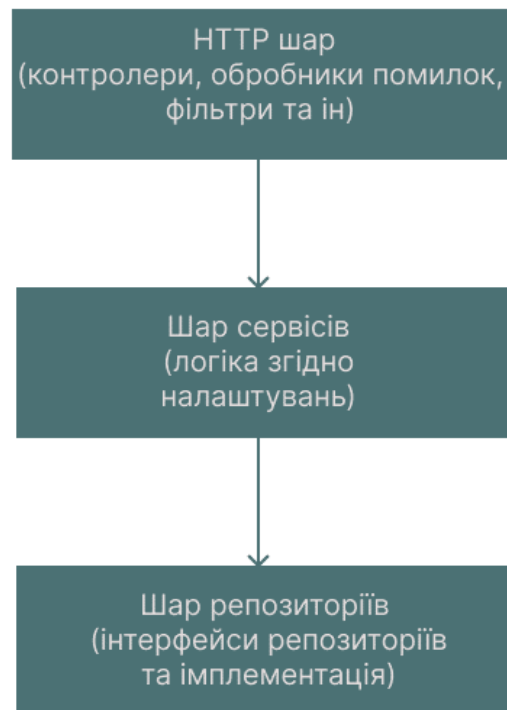


Рисунок 2.3 — Представлення шарів додатку

Описаний функціонал можна покрити трьома шарами (рисунок 2.3):

HTTP шар – шар, який є найвищим шаром веб-додатка. Він відповідає за обробку вхідних запитів користувача та повернення правильної відповіді клієнту.

Також повинен обробляти виключення, створені іншими шарами, бути відповідальним за автентифікацію та виступати в ролі першої лінії захисту від несанкціонованого доступу.

Шар сервісів знаходиться нижче HTTP шару та містить сервіси з бізнес логікою що закладено в налаштуваннях платформи BAF.

Шар репозиторіїв – найнижчий шар додатку. Необхідний для зв'язку зі сховищем даних.

Наступним етапом є проектування інтерфейсу кожного шару. Для цього необхідно вирішити кілька питань. DTO (data transfer object) - це об'єкт, що виступає в ролі контейнера для даних. Ці об'єкти необхідні для передачі даних між різними процесами та шарами додатку. Модель домену складається з наступних об'єктів: Entity (сутність) - це об'єкт, що представляє собою якусь сутність і не містить логіки. Value object - це опис властивостей сутності, який не має власного життєвого циклу або представлення. Життєвий цикл value object прив'язаний до життєвого циклу сутності.

Давайте розглянемо проектування кожного шару, використовуючи наведені поняття:

- HTTP-шар повинен обробляти лише DTO.
- шар сервісів приймає об'єкти DTO в якості параметрів методів. Він також може працювати з об'єктами моделі домену, але повертає лише DTO в HTTP-шар;
- шар репозиторіїв працює з сутностями;
- модель домену містить внутрішню модель додатку. Якщо відкрити цю модель, клієнти отримають надмірні дані в неоптимальній формі. Обгортки DTO приховують надмірні та приватні дані та надають зрозуміле API;
- у разі зміни моделі домену не обов'язково змінювати DTO, тому кінцевий клієнт отримає дані тієї ж форми, що і до зміни.

Далі розглянемо шаблони, що використовуються. Шаблон проектування - це архітектурне рішення, яке описує, як вирішуються часті задачі при розробці програмних систем чи додатків.

Фреймворк Spring реалізує декілька шаблонів:

- інверсія управління (IoC) - це основний принцип розробки Spring;
- фабрика - Spring використовує цей шаблон для створення об'єктів beans через посилання на контекст програми;
- сінглтон за замовчуванням - це конфігураційний файл (XML), який створюється лише один раз, незалежно від кількості викликів методу `getBean()`. Це через те, що всі bean в Spring за замовчуванням є сінглтонами;
- шаблон Model View Controller - основна мета застосування цієї концепції полягає в розділенні бізнес-логіки (моделі) від її візуалізації (представлення). Це забезпечує більшу можливість повторного використання та гнучкість системи.

2.6 Опис архітектури мобільного додатку

Для створення клієнтського додатку на платформі iOS та Android рекомендує використовувати архітектурний патерн MVC (Model-View-Controller), проте на практиці в контексті часто спостерігається тенденція до трансформації літери "M" з Model в "Massive", оскільки розділення View та Controller зазвичай не виконується, і логіка для екрану переноситься або в сам ViewController (що фактично є представленням у контексті MVC), або перевантажується моделлю.

Окрім MVC, широко використовуються архітектурні патерни MVVM (Model-View-ViewModel) та VIPER (clean architecture). На основі VIPER побудовані великі проекти із залежностями, в той час як MVVM вважається одним із найкращих підходів для середніх проектів[1].

MVVM спрощує відокремлення розробки графічного інтерфейсу від розробки бізнес-логіки (бек-енд логіки), відомої як модель. Модель представлення є частиною, що відповідає за перетворення даних для їх подальшого використання і підтримки. Таким чином, модель представлення більше схожа на модель, ніж на представлення, і обробляє більшість логіки відображення даних. Крім того, модель представлення може реалізовувати патерн "медіатор", організовуючи доступ до

бек-енд логіки навколо набору правил використання, які підтримуються представленням.

Шаблон MVVM складається з трьох частин: Моделі (Model), Представлення (View) та Моделі вигляду (ViewModel). Він має два основних недоліки: логіка переходів міститься в представленні (ViewController), що порушує принцип Single Responsibility та погіршує тестованість, і ViewModel має логіку створення модулів (View-ViewModel).

Для усунення цих недоліків необхідно виділити додаткові шари програми, що відповідатимуть за переходи та створення модулів. Таким чином, було прийнято рішення розробити додатковий клас Router, який міститиме методи з логікою переходів між модулями та статичний метод для створення поточного модуля. Посилання на цей клас буде зберігатися в ViewModel(рисунок 2.4).

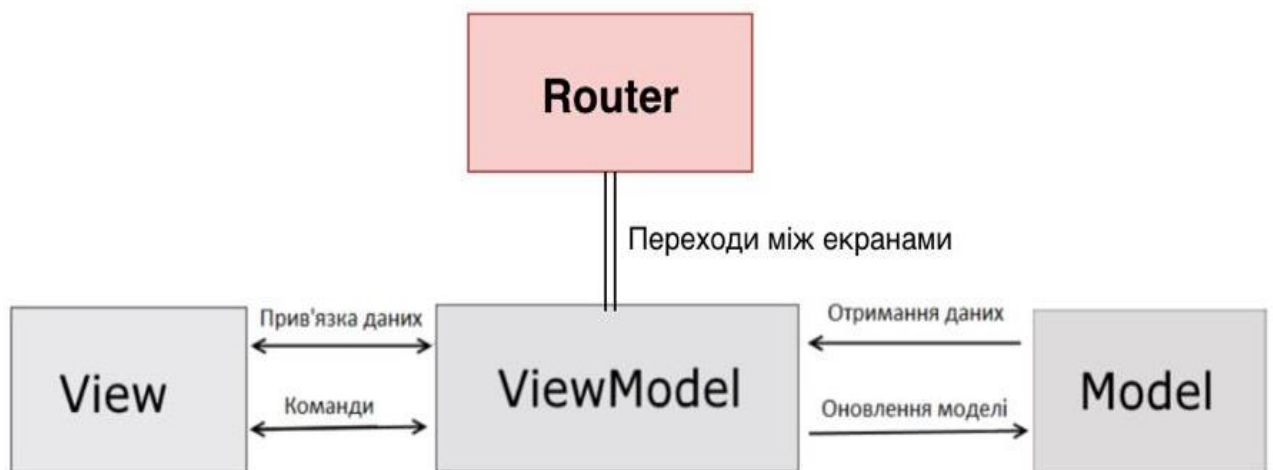


Рисунок 2.4 – Схема MVVMR шаблону

2.7 Опис інструментів розробки

Одним із популярних інструментів для крос-платформної розробки є платформа Xamarin, яка дозволяє створювати рідні мобільні додатки для операційних систем iOS, Android та Windows з єдиної кодової бази.

Xamarin надає розробникам широкий спектр інструментів, які забезпечують ефективну та продуктивну розробку, тестування та публікацію мобільних додатків[31].

Xamarin.Forms - це платформа інтерфейсу користувача з відкритим вихідним кодом. За допомогою Xamarin.Forms розробники можуть створювати додатки для Xamarin.Android, Xamarin.iOS та Windows на основі спільного коду.

Xamarin.Forms дозволяє розробникам створювати інтерфейси користувача у XAML за допомогою програмного коду на C#. Ці інтерфейси на кожній платформі готуються до перегляду як власні елементи керування.

Для кого призначена платформа Xamarin.Forms Платформа Xamarin.Forms спрямована на розробників, які мають такі завдання:

Спільне використання макету інтерфейсу користувача та розробка на різних платформах. Спільне використання коду, тестів та бізнес-логіки на різних платформах. Написання кросплатформених додатків мовою C# в середовищі Visual Studio.

Xamarin.Forms надає узгоджений API для створення елементів інтерфейсу користувача на різних платформах. Цей API може бути реалізований у XAML або C# і підтримує зв'язування даних для шаблонів, таких як "Модель-Вид-Модель представлення" (MVVM).

Під час виконання Xamarin.Forms використовує візуалізатори платформи для перетворення кросплатформених елементів інтерфейсу користувача на власні елементи керування в Xamarin.Android, Xamarin.iOS та UWP. Описаний функціонал роботи Xamarin (рисунок 2.5):

Це дозволяє розробникам досягти звичного вигляду, поведінки та продуктивності, а також використовувати переваги спільного використання коду на різних платформах.

Додатки Xamarin.Forms зазвичай складаються зі спільної бібліотеки NET Standard та окремих проектів платформ.

Спільна бібліотека містить представлення XAML або C# та весь бізнес-логіку, наприклад сервіси, моделі або інший код. Проекти платформи містять всю логіку, залежну від платформи або пакети, необхідні додатку.

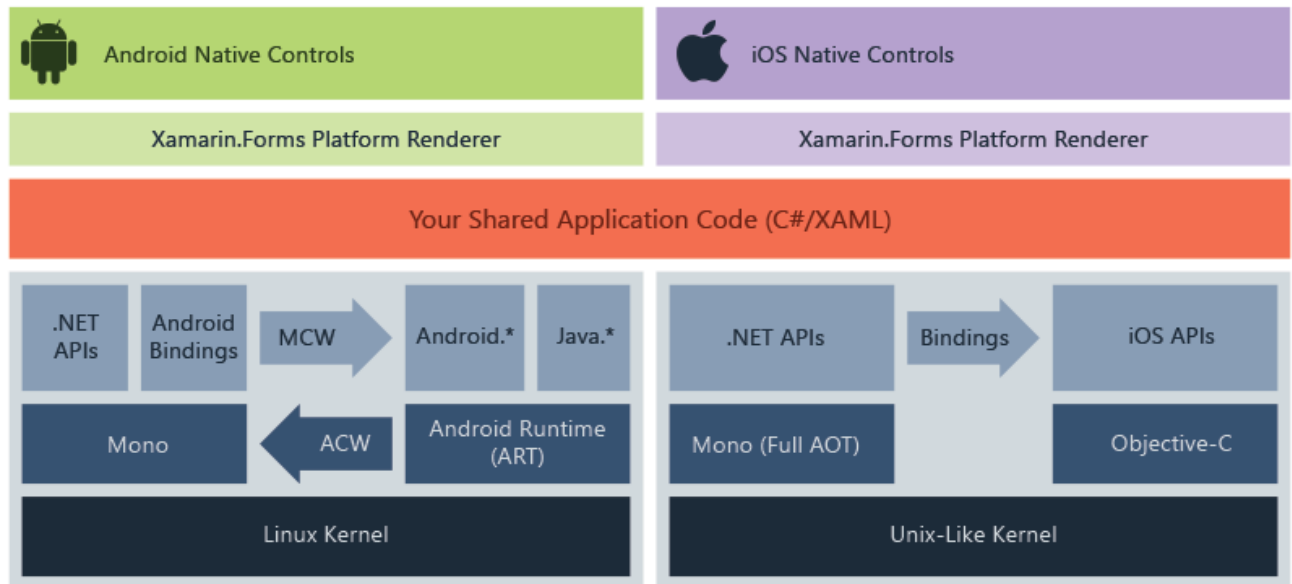


Рисунок 2.5 – Схема роботи Xamarin.Forms

Xamarin.Forms використовує платформу Xamarin для виконання додатків .NET на різних платформах за допомогою власних засобів.

Додаткові можливості Xamarin.Forms має велику екосистему бібліотек, які додають різноманітні функціональні можливості для додатків. У цьому розділі описані деякі з цих додаткових функцій.

Xamarin.Essentials - це бібліотека, яка надає кросплатформений API для службових програм пристрою. Як і сама платформа Xamarin, бібліотека Xamarin.Essentials є абстракцією, яка спрощує процес доступу до власних службових засобів.

2.8 Доступ до метаданих

Кожна система управління базами даних (СУБД) зберігає метадані про всі сутності бази даних. Наприклад, у SQL Server за допомогою інструкції CREATE

можна створити 52 різні сутності, такі як ASSEMBLY, INDEX, TABLE, FUNCTION, DATABASE тощо.

Хоча в різних СУБД використовуються різні назви для метаданих, наприклад, системний каталог чи словник даних, спільною рисою є те, що каталог або словник самі складаються з таблиць, а точніше - системних таблиць.

Це дозволяє користувачам звертатися до метаданих так само, як і до прикладних даних, використовуючи інструкцію SELECT. Зміни в каталозі/словнику відбуваються автоматично при виконанні відповідних інструкцій користувачем, що змінюють стан об'єктів бази даних. Системні таблиці не повинні змінюватися безпосередньо жодним користувачем.

Microsoft SQL Server надає низку колекцій системних представлень, що містять метадані:

- представлення інформаційної схеми, визначені в особливій схемі "INFORMATION_SCHEMA" і включають 20 представлень, таких як TABLES та VIEWS;
- представлення каталогів баз даних і файлів, які містять 6 представлень, наприклад, sys.database_mirroring, sys.database_recovery_status, sys.databases;
- метадані серверного рівня, рівня бази даних, інформацію про шифрування;
- системні збережені процедури, функції метаданих, конфігурації, безпеки, системні функції, команда DBCC;
- в платформі VAF при використанні публікації на веб сервері вони зберігаються в БД.

Ці метадані дозволяє ефективно керувати та обслуговувати бази даних SQL Server.

2.9 Висновки до розділу 2

Для ефективного проектування інтерфейсу було обрано інструмент Figma, який дозволяє створювати високоякісні та інтуїтивні макети з дотриманням

принципів адаптивності, юзабіліті, візуальної привабливості. Figma забезпечує можливості прототипування, спільної роботи та швидкого внесення змін.

Архітектура системи складається з бази даних MS SQL Server для надійного зберігання даних, CRM системи на BAF як центрального модуля, HTTP сервісів для безпечного обміну даними та мобільного додатку на Xamarin для забезпечення зручного інтерфейсу користувача.

Для серверної частини обрано архітектуру з трьома шарами: HTTP, сервісів та репозиторіїв. Використання DTO, патернів проектування Spring та відповідного розподілу відповідальностей забезпечує гнучкість, масштабованість та безпеку системи.

Для клієнтської частини рекомендовано використовувати архітектурний патерн MVVM з додатковим класом Router для покращення тестованості та дотримання принципу Single Responsibility.

Як інструмент крос-платформної розробки обрано Xamarin з використанням Xamarin.Forms для створення інтерфейсів з єдиної кодової бази на C# та XAML.

Застосування описаних засобів та технологій дозволяє розробити якісний мобільний додаток з сучасним інтерфейсом, надійною архітектурою та ефективними інструментами розробки, забезпечуючи високу продуктивність та задоволеність користувачів.

3 РОЗРОБКА МЕТОДУ ТА ОПТИМІЗАЦІЯ ЙОГО ЕФЕКТИВНОСТІ

3.1 Метод опису метаданих для платформи ВАF

Для підвищення продуктивності системи та ефективності роботи користувачів, потрібно реалізувати новий метод опису метаданих, який дозволить полегшити та прискорити роботу з великими об'ємами даних в системі. Підвищення продуктивності передбачається на основі реалізації нового методу суть якого полягає у створенні Http сервісу в платформі ВАF. З назвою MobileHTTP (рисунок 3.1).

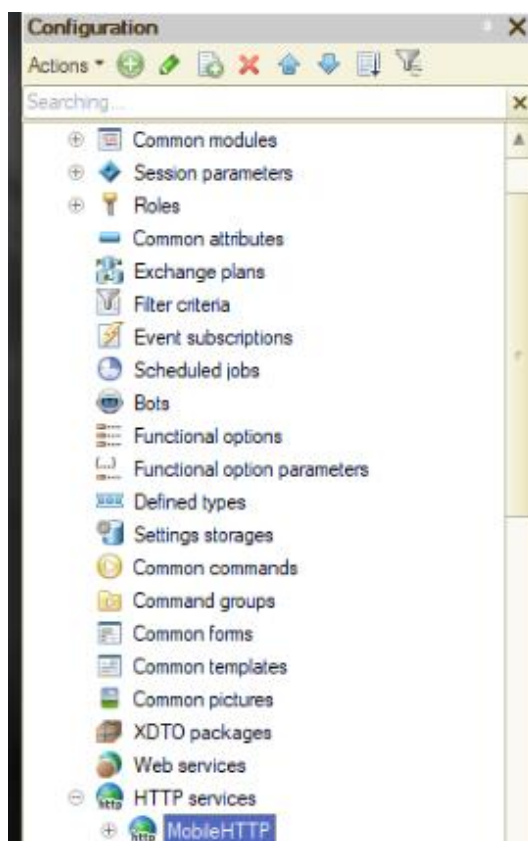


Рисунок 3.1 – Додавання об'єкту в конфігурацію

Для роботи HTTP-сервісу в платформі потрібно налаштувати відповідні параметри. Зазвичай ці налаштування зберігаються в довіднику платформи ВАF.

Довідник (Catalogs) - це спеціальний компонент платформи, який дозволяє зберігати та управляти різноманітними налаштуваннями та довідковою

інформацією, необхідною для функціонування додатків, побудованих на платформі VAF.

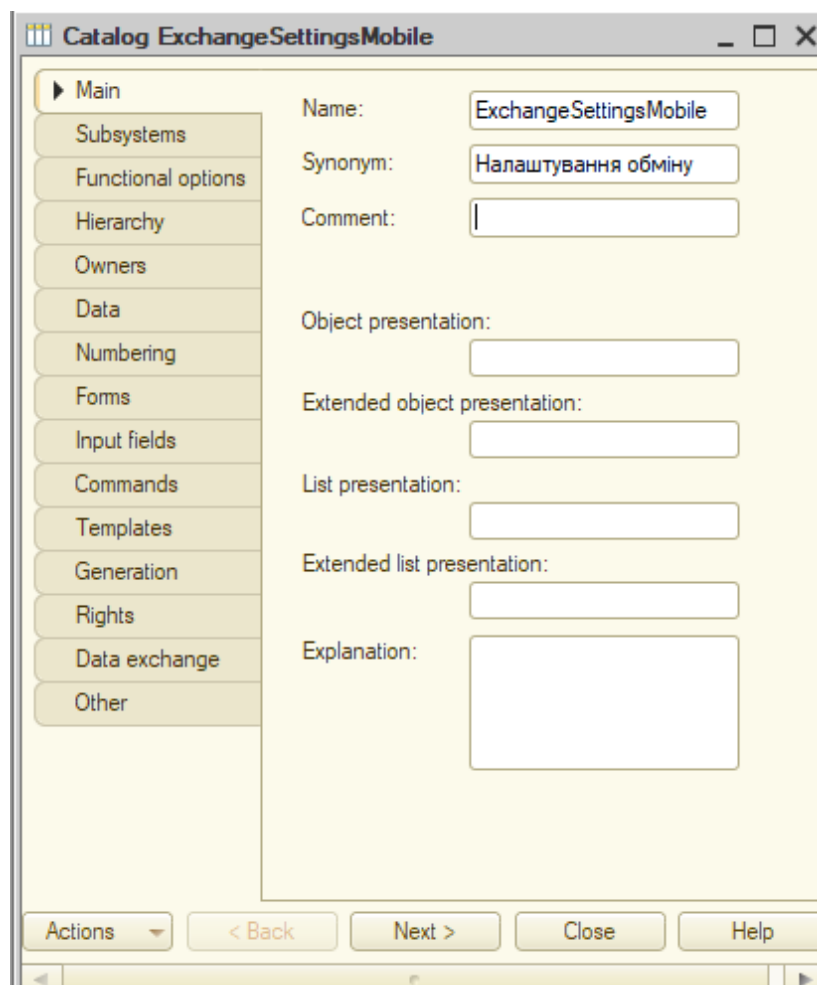


Рисунок 3.2 – Додавання довідника налаштувань в конфігурацію

Довідник Settings Exchange Mobile (рисунок 3.2), який використовується для налаштування HTTP-сервісу в платформі VAF, має ієрархічну структуру з 2 рівнями, то його можна представити наступним чином:

1. Верхній рівень (1-й рівень): Цей рівень містить загальні налаштування і використовується в якості групи. Наприклад стандартні об'єкти метаданих документи, довідники або задачі. Структуру довідника подано на рисунку 3.3.

2. Нижній рівень (2-й рівень): Цей рівень містить налаштування, специфічні для конкретного об'єкту. Наприклад якщо підкаталог «Документ» то дочірні елементи.



Рисунок 3.3 – Ієрархічної структура довідника «Settings Exchange Mobile»

Для встановлення цих налаштувань потрібно перейти в вкладку Hierarchy та встановити значення що довідник є ієрархічний як показано на рисунку 3.4.

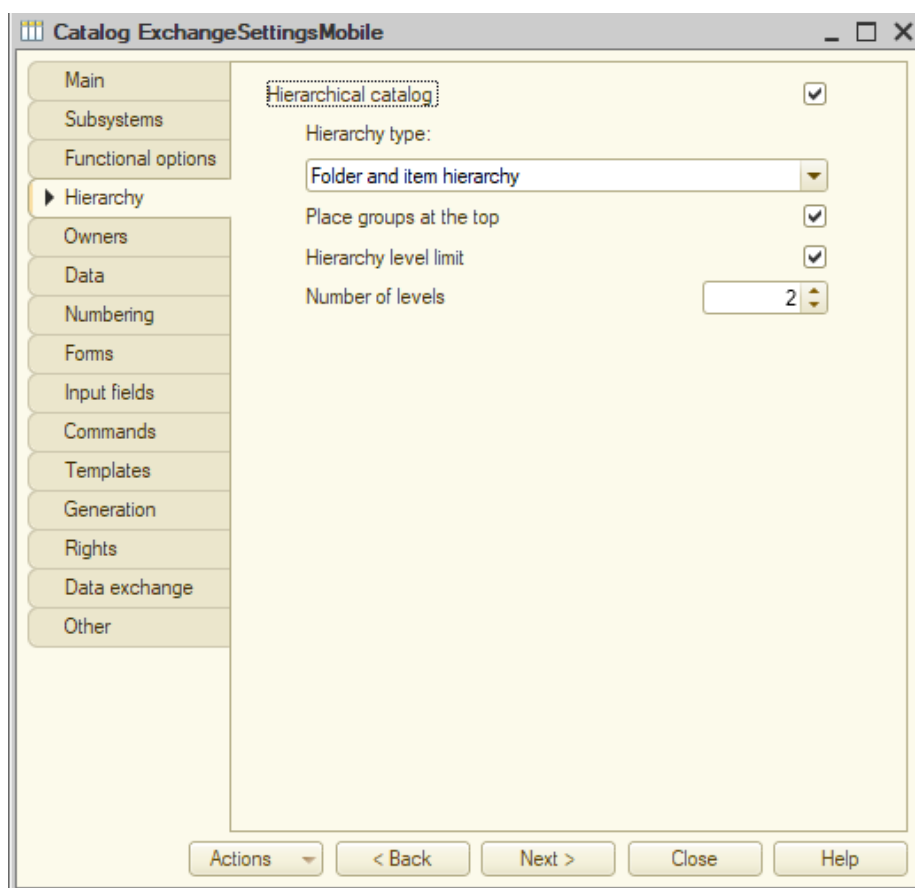


Рисунок 3.4 – Встановлення ієрархічної структури довідника

У довіднику зберігаються поля (attributes) що наведено у таблиці 3.1.

Таблиця 3.1 – Поля поля (attributes) довідника

Назва	Тип	Призначення
Object	String	Містить назву об'єкта метаданих який буде описано в довіднику
Order	String	Порядок в списку
Synonym	String	Назва об'єкта метаданих яку бачить користувач у мобільному додатку
Service	Bool	Показник що об'єкт є службовий
Image	String	Назва картинки. Повний перелік доступних картинок по замовчуванню подано в Додатку Б.
UseRandom Request	Bool	Признак використання додаткового методу для виведення даних. Потрібний в особливих та спецефічних випадках
ArbitraryRequestMethod	String	Мість код який буде виконано в методі «Execute»

Для опису всіх необхідних реквізитів об'єкта потрібна таблиця. В ній будуть визначені всі налаштування полів. Дані наведено у таблиці 3.2.

Таблиця 3.2 – Поля (attributes) таблиці TabReports довідника

Назва	Тип	Призначення
Name	String	Містить назву об'єкта метаданих який буде описано в довіднику
Synonym	String	Порядок в списку
View	Bool	Перегляд
Editing	Bool	Редагування

Продовження таблиці 3.2 – Поля (attributes) таблиці TabReports довідника

ElementType	Ref. TypesofElements	Вид поля, на формі елемента.
Official	Bool	Признак використання поля для представлення об'єкту
Types	String	Вказує тип поля
OrderInList	int	Порядок в списку
OrderFilterList	int	Порядок в списку фільтру
OrderInFormElement	int	Порядок в формі елемента

Для опису всіх необхідних реквізитів таблиць об'єкта необхідна таблиця TabParts. В ній будуть визначені всі налаштування полів. Дані наведено у таблиці 3.3.

Таблиця 3.3 – Поля (attributes) таблиці TabParts довідника

Назва	Тип	Призначення
Name	String	Містить назву об'єкта метаданих який буде описано в довіднику
Synonym	String	Порядок в списку
View	Bool	Назва об'єкта метаданих яку бачить користувач у мобільному додатку
Editing	Bool	Показник що об'єкт є службовий
Element Type	Ref	Назва картинки
Official	Bool	Признак використання додаткового методу для виведення даних. Потрібний в особливих та спецефічних випадках
Types	String	Код який буде виконано в методі «Execute»

Для опису всіх команд та кнопок доступних користувачу потрібна таблиця TabCommands. В ній будуть визначені команди різних типів для об'єкта. Дані наведено у таблиці 3.4.

Таблиця 3.4 – Поля (attributes) таблиці TabCommands довідника

Назва	Тип	Призначення
Order	int	Порядок
Command	String	Команда (кнопка) що буде використовуватися в списку елементів

Для опису всіх необхідних таблиць об'єкта необхідна додати таблицю TabTables. В ній будуть визначені всі таблиці що використовує об'єкт. Дані наведено у таблиці 3.5.

Таблиця 3.5 – Поля (attributes) таблиці TabTables довідника

Назва	Тип	Призначення
Name	String	Містить назву об'єкта метаданих який буде описано в довіднику
Synonym	String	Порядок в списку
Order	Bool	Назва об'єкта метаданих яку бачить користувач у мобільному додатку
Image	Bool	Показник що об'єкт є службовий

Використання довідника для зберігання налаштувань об'єктів конфігурації дозволяє централізовано управляти цими параметрами та забезпечує гнучкість у конфігурації системи.

Для опису команд (кнопок) необхідно додати довідник Commands. Він буде містити дані про команди, які зможе натискати користувач в мобільному додатку.

Таблиця 3.6 – Поля (attributes) таблиці TabTables довідника

Назва	Призначення
AddNewCommand	Створення елемента, документа, задачі і тд.
DeleteCommand	Видалення елемента, документа, задачі і тд.
EditCommand	Редагування елемента, документа, задачі і тд.
FilterCommand	Встановлення фільтру
SaveNewCommand	Збереження нового елемента
GenerateReportCommand	Формування звіту
SavePDF	Збереження у файл
AgreeCommand	Підтвердження дії
RejectCommand	Скасування дії

Використання довідника для зберігання команд кожного об'єкта, дозволяє управляти цими ними, що забезпечує гнучкість у конфігурації системи.

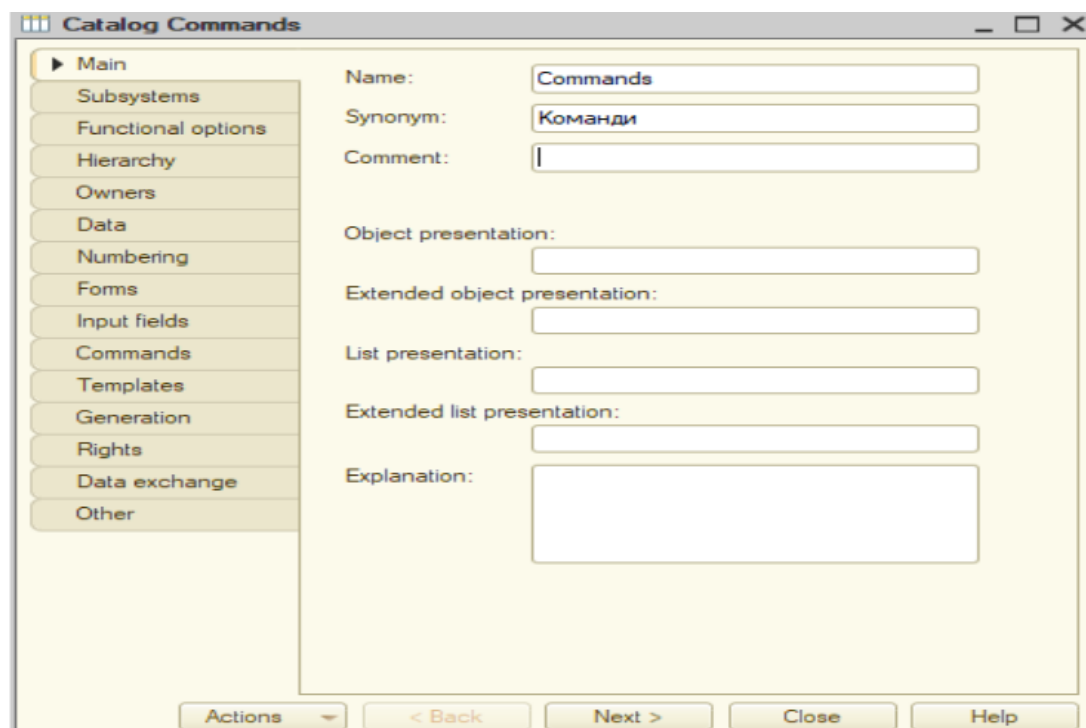


Рисунок 3.5 – Створення довідника Commands

Також для роботи довідника Commands необхідні поля (attributes) що наведено у таблиці 3.7.

Таблиця 3.7 – Поля (attributes) довідника Commands

Назва	Тип	Призначення
Synonym	String	Містить назву об'єкта метаданих який буде описано в довіднику
Purpose	Ref. TeamAssignments	Порядок в списку
Event	String	Назва об'єкта метаданих яку бачить користувач у мобільному додатку
Image	String	Показник що об'єкт є службовий

Команди мають також призначення вони можуть застосовуватися для різних варіантів відкриття об'єкту (в списку, формі вибору та об'єкті). Тому потрібно додати перелічення TeamAssignments. Його значення подано в таблиці 3.8

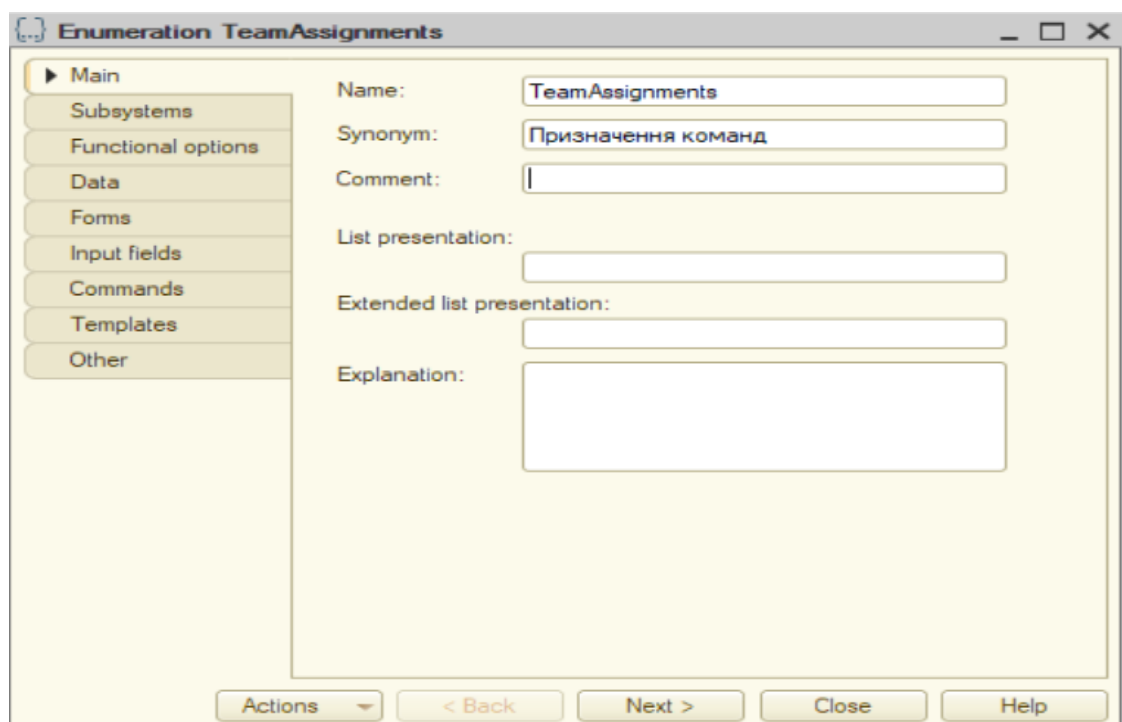


Рисунок 3.6 – Створення довідника TeamAssignments

Таблиця 3.8 – значення перелічення TeamAssignments

Назва	Призначення
List	Використовується на формі списку
Element	Використовується на формі об'єкта
NewElement	Використовується на формі нового об'єкта, тобто у випадку коли ще не створено посилання

Кожне поле повинно мати своє значення, та відображати дані для користувачів. Поля мають різне призначення і вони можуть застосовуватися для різних варіантах форм. Його значення подано в таблиці 3.9.

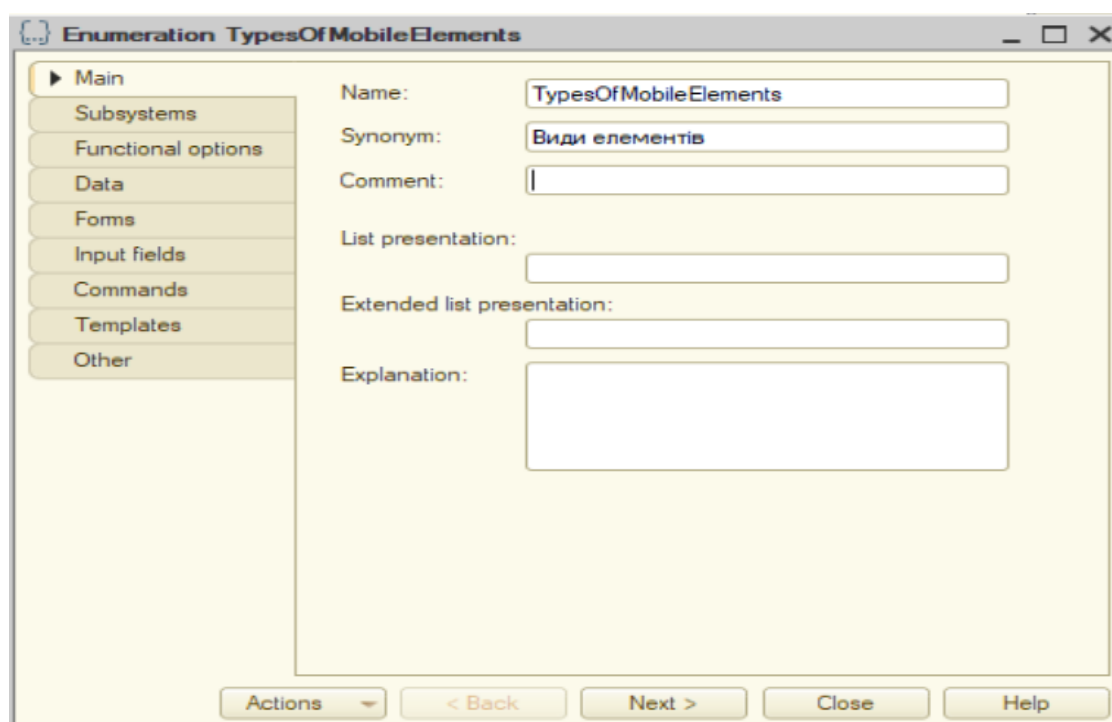


Рисунок 3.7 – Створення довідника TypesOfMobileElements

Таблиця 3.9 – значення перелічення TypesOfMobileElements

Назва	Призначення
Entry field	Поле введення
InputFieldEmail	Поле E-mail
InputFieldNumeric	Поле цифр

Продовження таблиці 3.9 – Поля (attributes) таблиці TabReports довідника

InputFieldTelephone	Поле телефону
InputFieldUri	Поле посилання
Switch	Перемикач
FieldLabel	Поле напису
DateInputField	Поле для дати
InputFieldLink	Поле для посилання
InputFieldDateTime	Поле для вибору дати та часу
InputFieldEnumeration	Поле для вибору перелічення

Для ефективного збереження історії авторизації важливо забезпечити надійне, захищене та довготривале зберігання цих даних, а також забезпечити можливість їх аналізу та звітування. Це дозволить підвищити рівень безпеки, прозорості та підзвітності в управлінні доступом до критично важливих ресурсів. А саме, активності мобільного додатку.

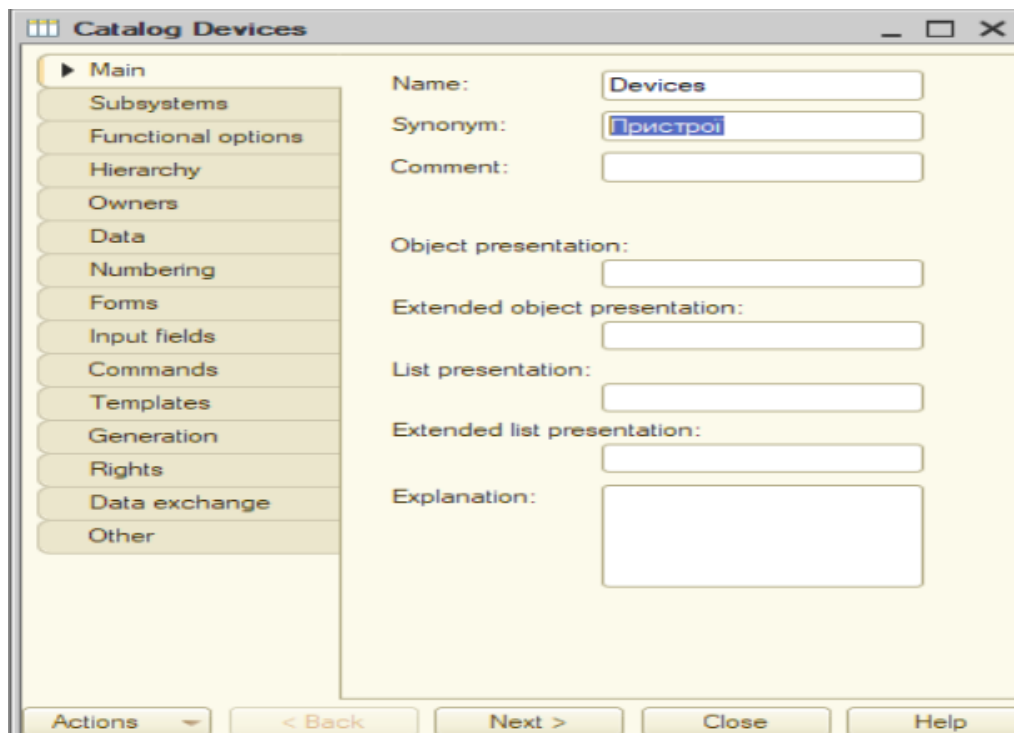


Рисунок 3.8 – Створення довідника Devices

Тому потрібно створити довідник Devices. Він буде описувати пристрої які використовують користувачі. В таблиці 3.10 наведено реквізити довідника

Таблиця 3.10 – Поля (attributes) довідника Devices

Назва	Тип	Призначення
Model	Строка	Модель пристрою
Manufacturer	Строка	Виробник
Version	Строка	Версія
Type	Строка	Тип пристрою
Platform	Строка	Платформа пристрою IOS або Android
ID	Строка	Ідентифікатор пристрою є унікальним
User	Ref. User	Службове поле в якому зберігається прикріплений користувач

Записи про кожну авторизацію та вихід з додатку будуть записуватися в не періодичний реєстр відомостей MobileAppJournalAuthorization. В таблиці 3.11 наведено виміри та ресурси реєстра.

Рисунок 3.8 – Створення довідника Devices

Таблиця 3.11 – Ресурси та виміри реєстра відомостей MobileAppJournalAuthorization

Назва	Ресурс/Вимір	Тип	Призначення
Device	Вимір	Ref. Devices	Пристрій з якого здійснено запис
Date	Вимір	Дата	Виробник
User	Ресурс	Ref. User	Службове поле в якому зберігається користувач
Type	Вимір	Ref	Зберігає тип входу в моб. додаток. Доступні значення: Entry, Exit , Login Re-entry

Блокування записів є важливим механізмом для запобігання конфліктів та забезпечення цілісності даних в системах управління доступом. Ось кілька ключових причин, чому блокування записів має важливе значення:

1. Уникнення перезапису даних: Без блокування кілька користувачів або процесів можуть одночасно намагатися оновити один і той самий запис, що призводить до втрати даних або некоректного оновлення.

2. Забезпечення узгодженості даних: Блокування гарантує, що оновлення записів здійснюються атомарно, тобто повністю або не здійснюються взагалі. Це запобігає появі проміжних, неузгоджених станів даних.

3. Підтримка транзакцій: У складних системах управління доступом блокування використовується для реалізації транзакційної обробки, коли кілька пов'язаних оновлень виконуються як єдина неподільна операція.

4. Координація паралельних операцій: Блокування дозволяє синхронізувати паралельні оновлення, запобігаючи гонці даних та конфліктам.

5. Забезпечення послідовності: Блокування гарантує, що оновлення записів виконуються в потрібному порядку, підтримуючи логічну послідовність змін.

Для реалізації цього функціоналу потрібно створити ще один реєстр відомостей BlockedObjectsUsers в ньому будуть зберігатися всі заблоковані об'єкти. Блокування буде відбуватися коли користувач відкриває об'єкт і в нього є доступ на його редагування. Після закриття блокування знімається. В таблиці 3.12 наведено виміри та ресурси реєстра.

Таблиця 3.12 – Ресурси та виміри реєстра відомостей BlockedObjectsUsers

Назва	Ресурс/Вимір	Тип	Призначення
Object	Вимір	Ref. Any	Модель пристрою
User	Вимір	Ref. Users	Службове поле в якому зберігається користувач
DateBlock	Ресурс	Date	Містить дату та час коли відбулося блокування

Рисунок 3.9 – Створення довідника Devices

Розглянемо структуру роботи HTTP сервісу в поєднанні з платформою ВАФ, яка використовує раніше створені елементи в його роботі, обробляючи запити

мобільного застосунку. Це структурна схема на рисунку 3.10, яка показує, як різні компоненти платформи взаємодіють між собою.

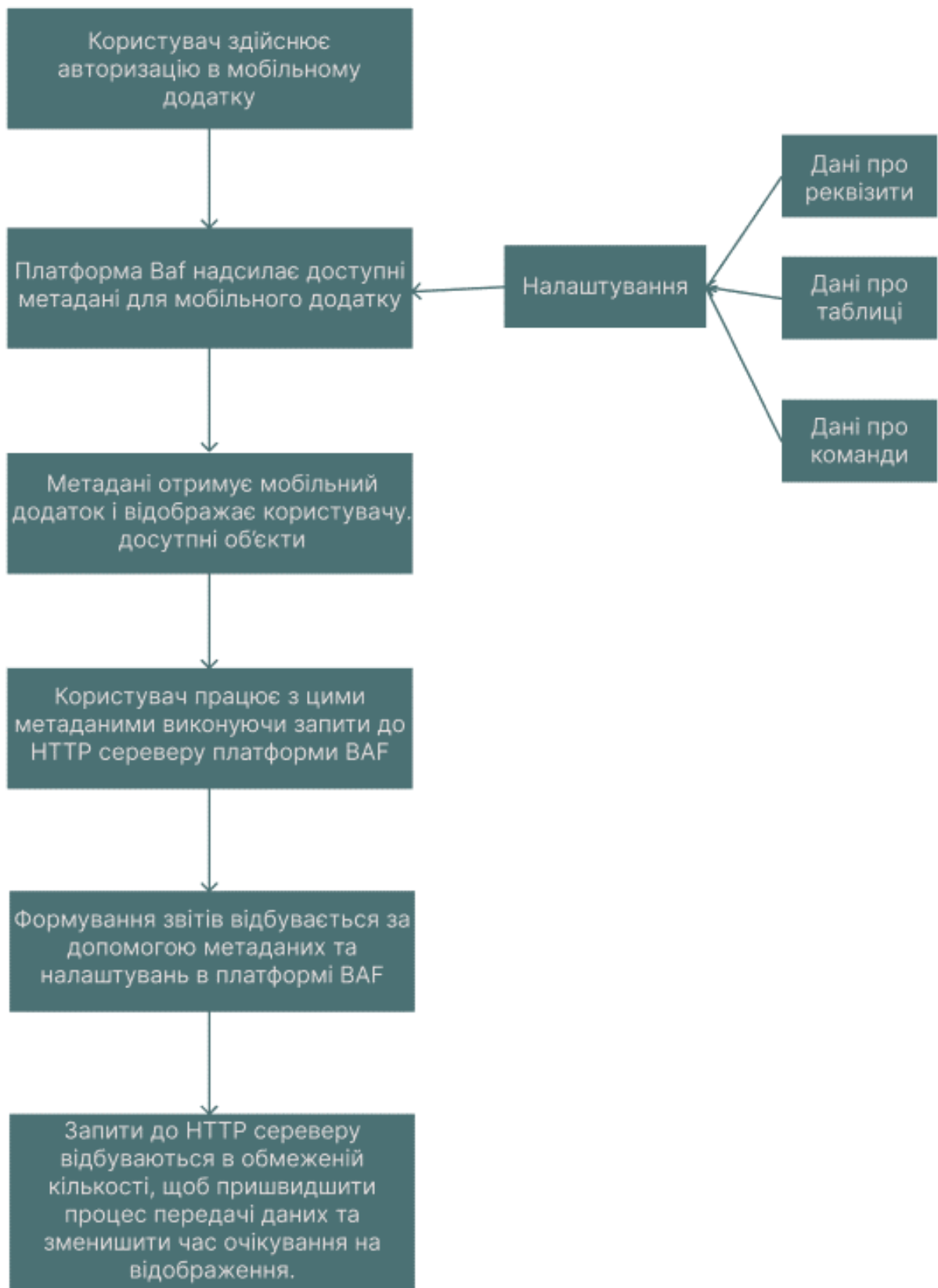


Рисунок 3.10 – Структура роботи методу

3.2 Оптимізація використання HTTP сервісів та протоколу OData

Для підвищення потенціалу платформи BAF актуальним стає створення надійного механізму інтеграції з веб-сервісами платформи Microsoft .NET. Така інтеграція дозволить розширити функціональність програм та забезпечити універсальний доступ до бази даних BAF.

Існують різні підходи до інтеграції, такі як OLE Automation, обмін файлами або використання зовнішнього компонента, що розширює вбудовану мову. Кожен з цих підходів має свої переваги та недоліки, які необхідно враховувати при виборі оптимального рішення.

Наприклад, підхід OLE Automation надає повний доступ до об'єктів BAF, але вимагає складного програмування та налагодження бізнес-логіки. Крім того, він обмежує продуктивність веб-сервісу, оскільки OLE-об'єкт працює на тому ж комп'ютері, де запущено веб-сервіс, і при аварійному зупиненні веб-сервісу в пам'яті зберігається об'єкт BAF, що вимагає розробки спеціальних алгоритмів для його примусового вивантаження.

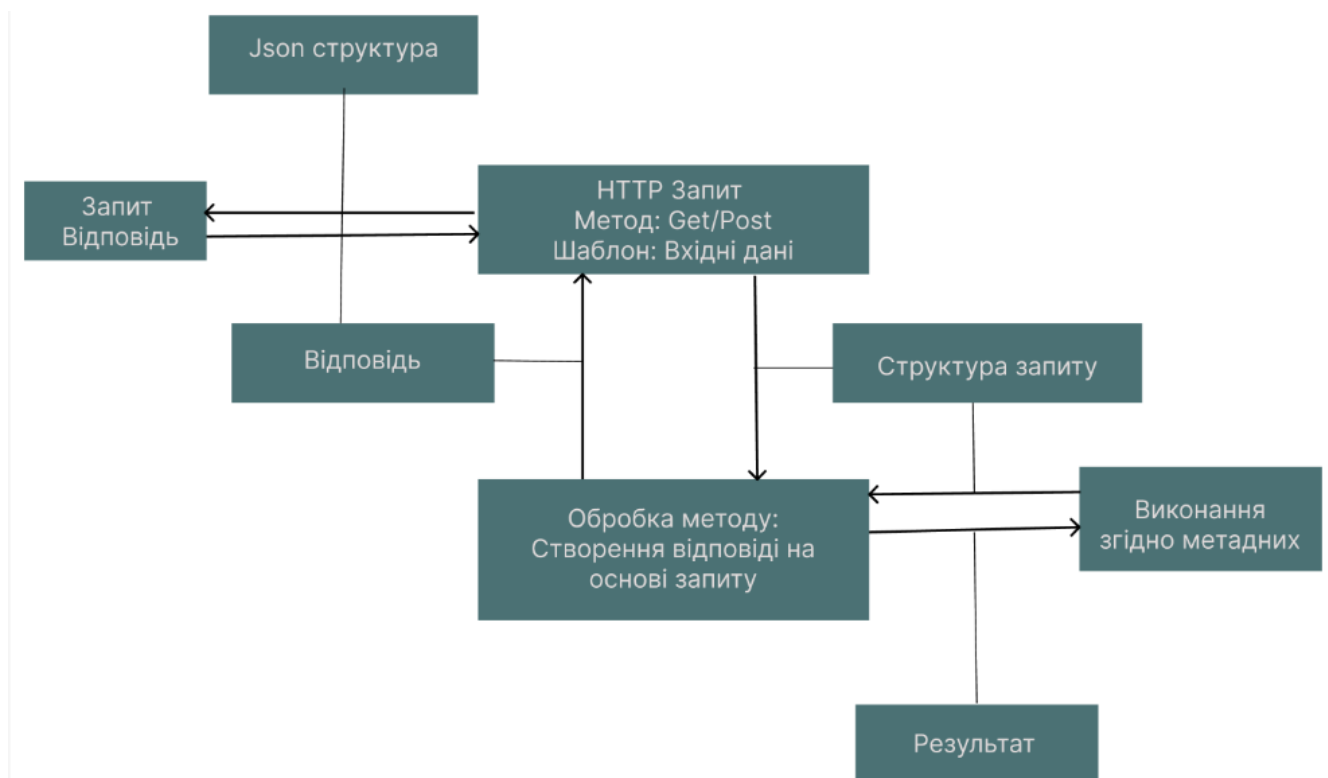


Рисунок 3.11 – Блок-схема роботи HTTP сервісів платформи BAF

Шаблон задає шлях, за яким може відбуватися звернення до HTTP-сервісу. У шаблоні можна використовувати визначений набір символів, у тому числі параметризовані сегменти виду.

Для кожного методу показано, по-перше, оброблюваний метод HTTP, а також створюється процедура на вбудованій мові, яка і буде виконувати обробку даних. Також можна вказати, що буде оброблятися не який-то конкретний, а будь-який HTTP-метод із доступних. На рисунку 3.11 подано блок схему роботи її HTTP сервісів.

В створеному раніше HTTP сервісу потрібно створити методи які забезпечать надійну, безпечну та швидку роботу мобільного застосунку. Для цього буде створено HTTP метод для авторизації Login, скріншот налаштувань при створенні методу подано на рисунку 3.12 але вона повинна бути безпечною. Для цього буде застосовано вбудований інструмент безпечної авторизації.

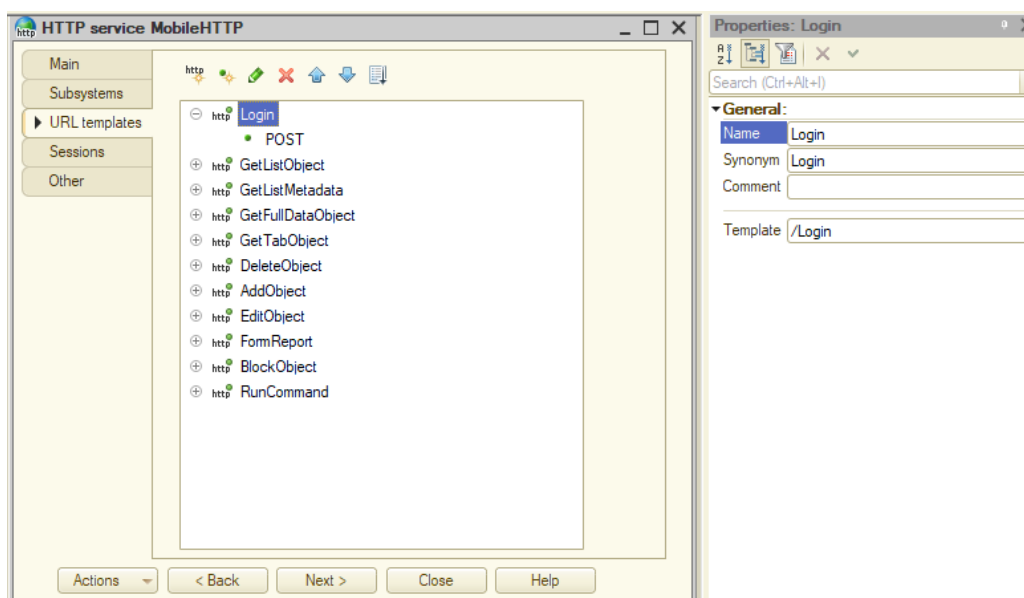


Рисунок 3.12 – Налаштування методу авторизації Login

Паролі для більшості сайтів передаються у відкритому вигляді, якщо не використовується SSL. Відмова від використання SSL пов'язана в більшості випадків з адміністративним фактором. Тим не менш, проблему передачі паролів і

логінів у відкритому вигляді треба вирішувати. Це можна зробити за допомогою вбудованої функції Безпечна авторизація.

Безпечна авторизація - функція, що забезпечує зашифровану передачу паролю користувача. Паролі користувачів хешуються за алгоритмом SHA-512, але шифрування паролю не є заміною SSL. Безпечна авторизація захищає від перехоплення паролю лише при прослуховуванні трафіку. Вимагає додаткових підключених бібліотек.

Якщо у зловмисника є можливість змінювати трафік, то шифрування не допоможе. При розробці функції виходили з того, що змінити трафік складніше, ніж прослухати його (той самий відкритий WiFi). Більше того, шифрування паролів не захищає від перехоплення сесії, але у сесії є свої механізми захисту (прив'язка до IP, часта зміна коду сесії). Загалом можна сказати, що рівень захисту підвищився, але краще використовувати SSL.

Безпечна авторизація працює в компоненті system.auth.authorize, при вході в Адміністративну частину системи цей механізм захисту не вмикається. Тому, все ж таки, постачальник рекомендує налаштування HTTP з'єднань.

Генерація ключа виконується один раз. Повторну генерацію слід проводити, якщо є підозри в компрометації ключа.

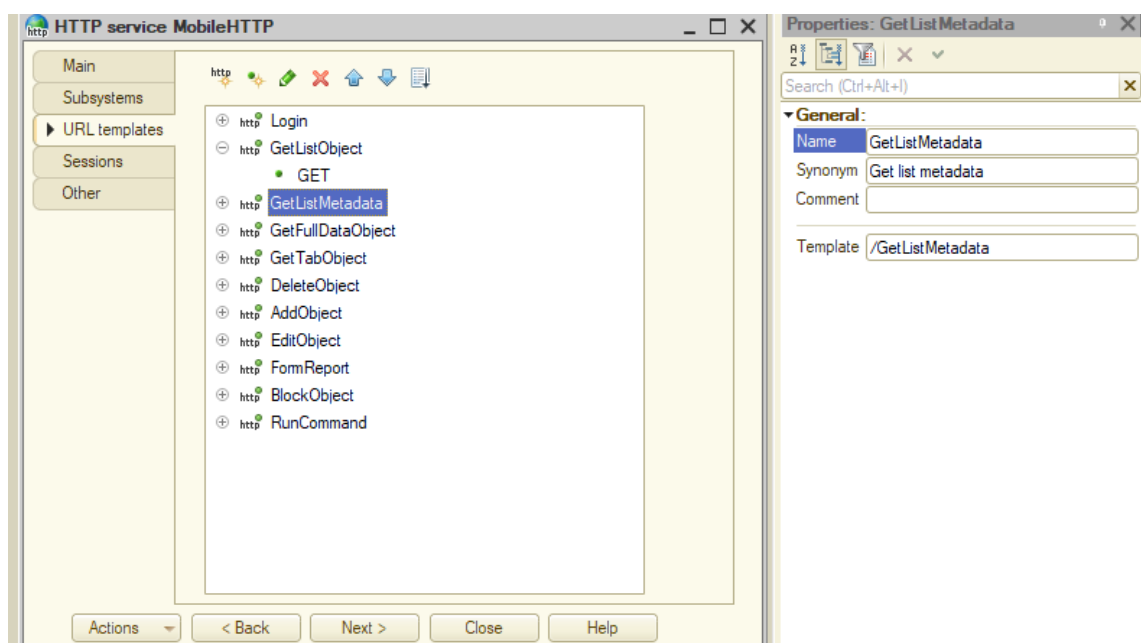


Рисунок 3.13 – Налаштування методу GetListMetadata

Після виконання успішної авторизації. В мобільний додаток має бути надіслано метадані про ті об'єкти з якими буде взаємодіяти користувач. За допомогою методу `GetListMetadata`. Метод буде виконувати перевірку по правах RLS і просто об'єктів які бачитиме користувач без можливості редагувати їх в подальшому тобто в режимі `Read only`. Скріншот налаштувань при створенні методу подано на рисунку 3.13.

Користувачу будуть відображені усі доступні об'єкти які визначено в налаштуваннях. Для отримання даних по певному об'єкту та виведенню їх у список, буде створено новий метод `GetListObject`. Зображення налаштувань при створенні методу подано на рисунку 3.14.

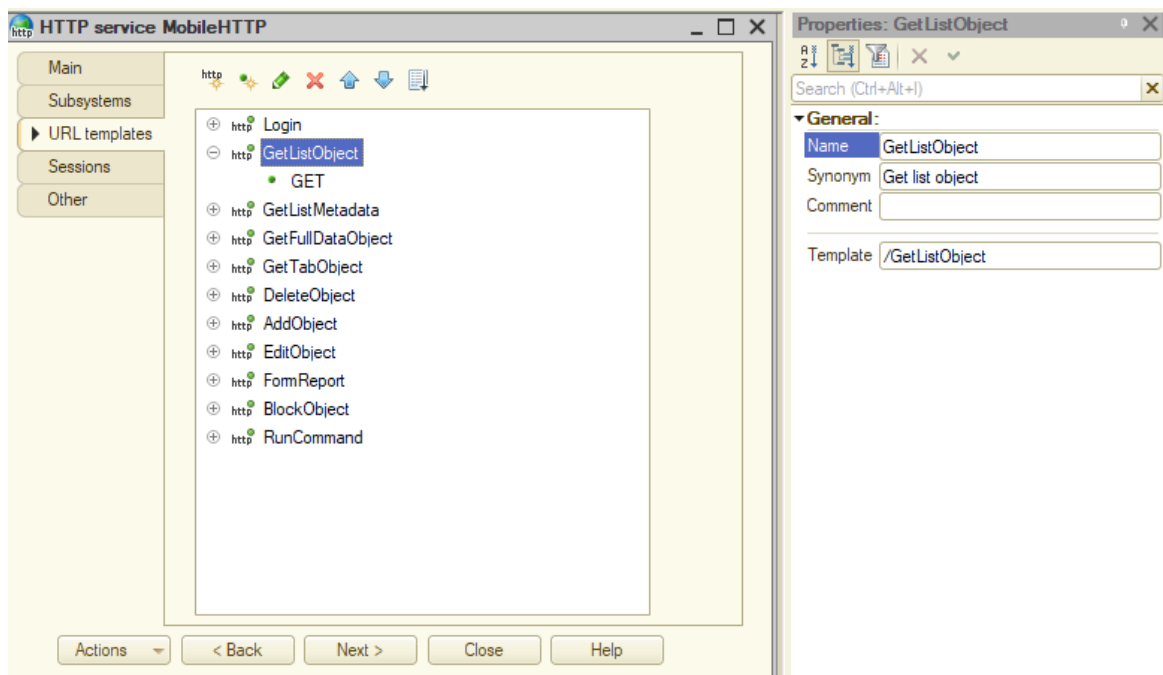


Рисунок 3.14 – Налаштування методу `GetListObject`

Після вибору зі списку певного об'єкта потрібно отримати всі данні по цьому об'єкту згідно налаштувань. Для цього буде створено метод `GetFullDataObject`. Він формуватиме дані по унікальному ідентифікатору певного об'єкта. Виводячи їх з можливістю редагувати або читати. Зображення налаштувань при створенні методу подано на рисунку 3.15.

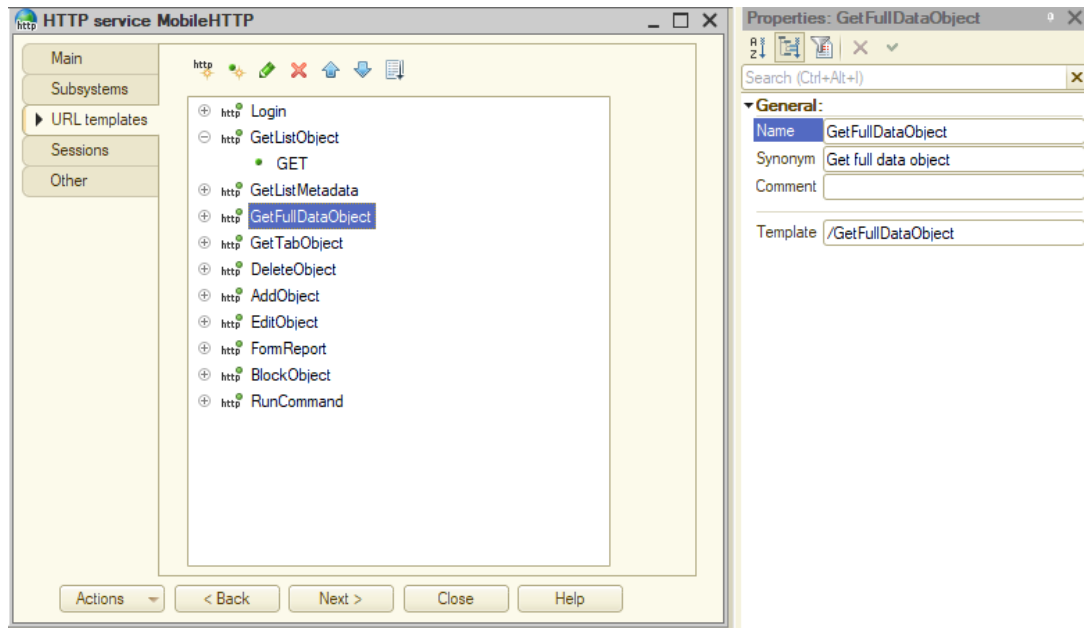


Рисунок 3.15 – Налаштування методу GetFullDataObject

Для об'єкта також потрібно відобразити не лише його поля, а й таблиці. Для цього в налаштуваннях довідника Settings Exchange Mobile, потрібно вказати які таблиці будуть відображатися. Та визначити налаштування для кожного поля. Щоб опрацьовувати метадані таблиць і відобразити таблиці з інформацією користувачу в мобільному додатку, буде створено метод GetTabObject. Зображення його налаштувань при створенні методу подано на рисунку 3.16.

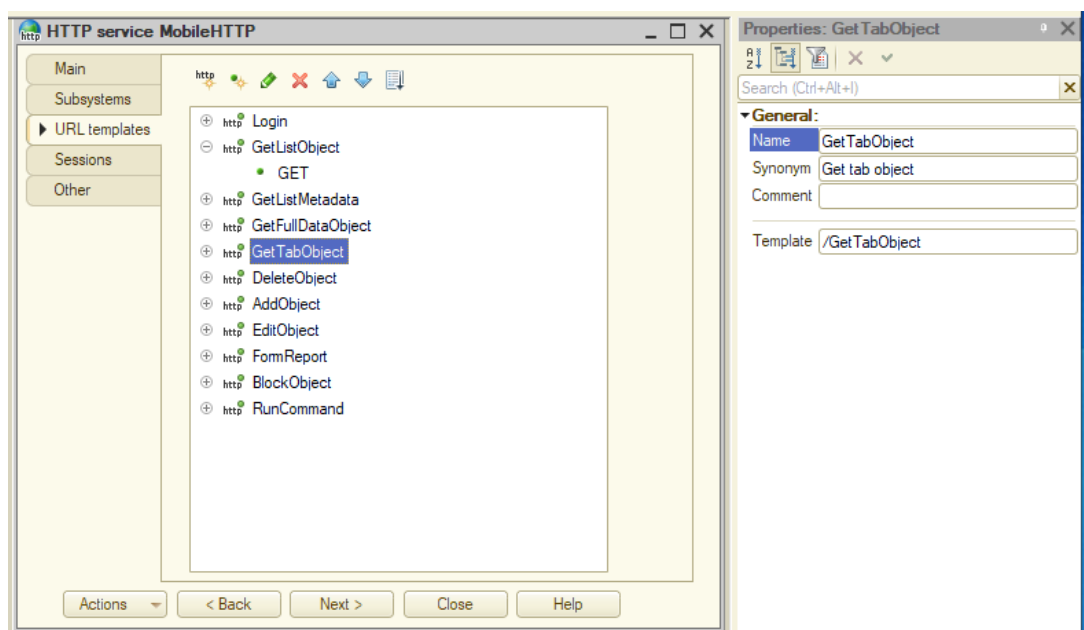


Рисунок 3.16 – Налаштування методу GetTabObject

Якщо у користувача є доступ на редагування об'єкту в нього буде можливість видалити цей об'єкт. Це буде реалізовано за допомогою команди Delete з довідника Commands. Команда обов'язково має бути додана до переліку команд у довіднику Settings Exchange Mobile. Нового метод DeleteObject який видалятиме його по унікальному ідентифікатору та метаданих об'єкта. Зображення налаштувань при створенні методу подано на рисунку 3.17.

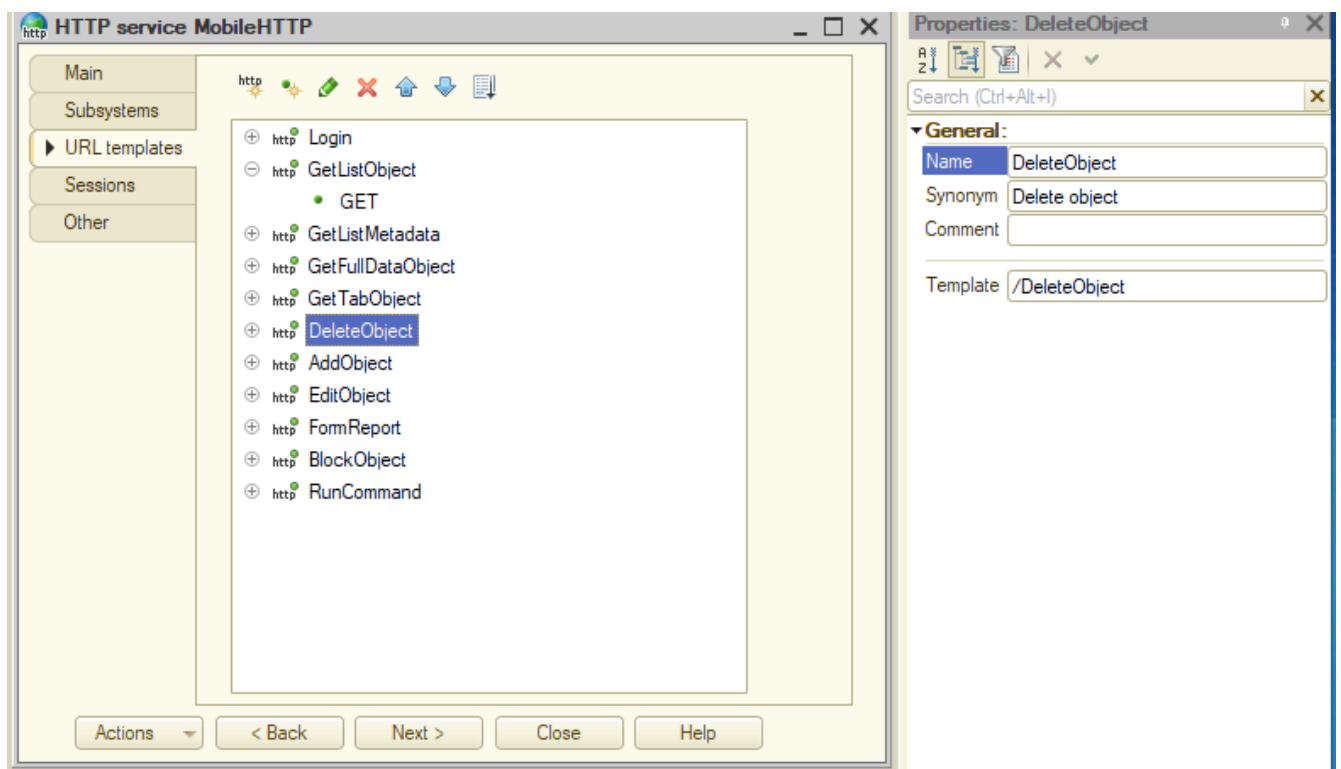


Рисунок 3.17 – Налаштування методу DeleteObject

У випадку якщо користувач створить новий об'єкт через команду буде створено пустий об'єкт. Для подальшого його збереження додано метод HTTP сервісу AddObject. Через який буде додано об'єкт в БД. Також в процесі запису спрацюють всі події які визначено в платформі VAF. З У випадку якщо користувач створить новий об'єкт через команду буде створено пустий об'єкт. Для подальшого його збереження додано метод HTTP сервісу AddObject. Зображення його налаштувань подано на рисунку 3.18.

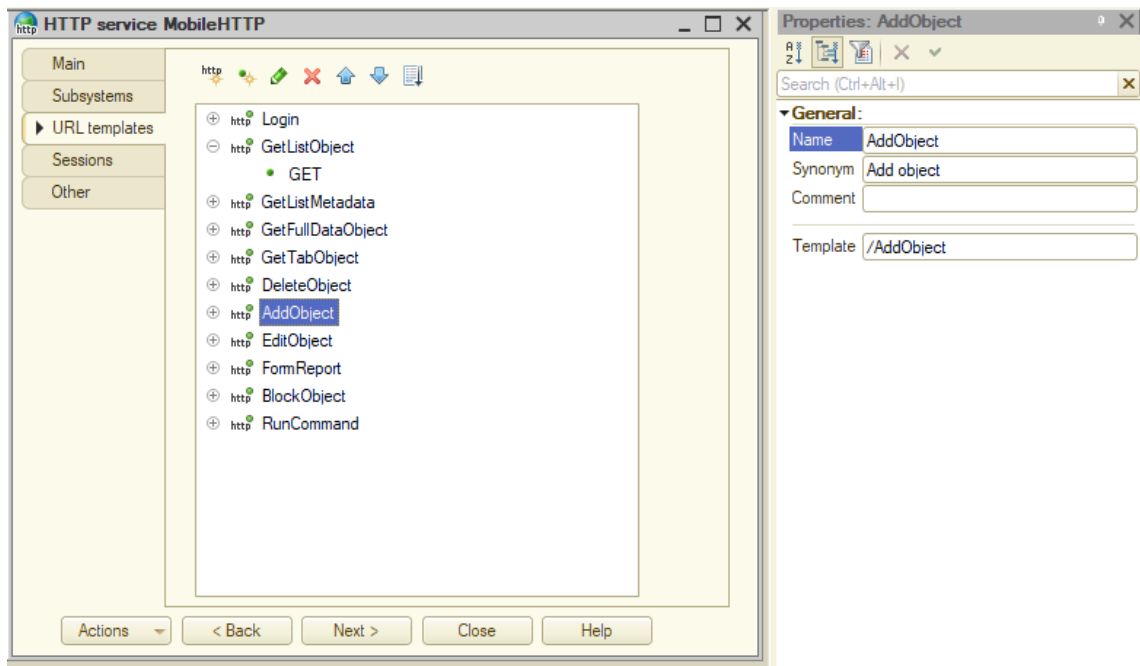


Рисунок 3.18 – Налаштування методу AddObject

Коли користувач буде редагувати новий об'єкт через команду або ж просто натисканням зі списку. Буде відредаговано вже існуючий об'єкт. Для виконання цієї дії буде додано метод HTTP сервісу EditObject. Через який буде редагувати об'єкт в БД. Також в процесі запису спрацюють всі події які визначено в платформі ВАР. Зображення його налаштувань подано на рисунку 3.19.

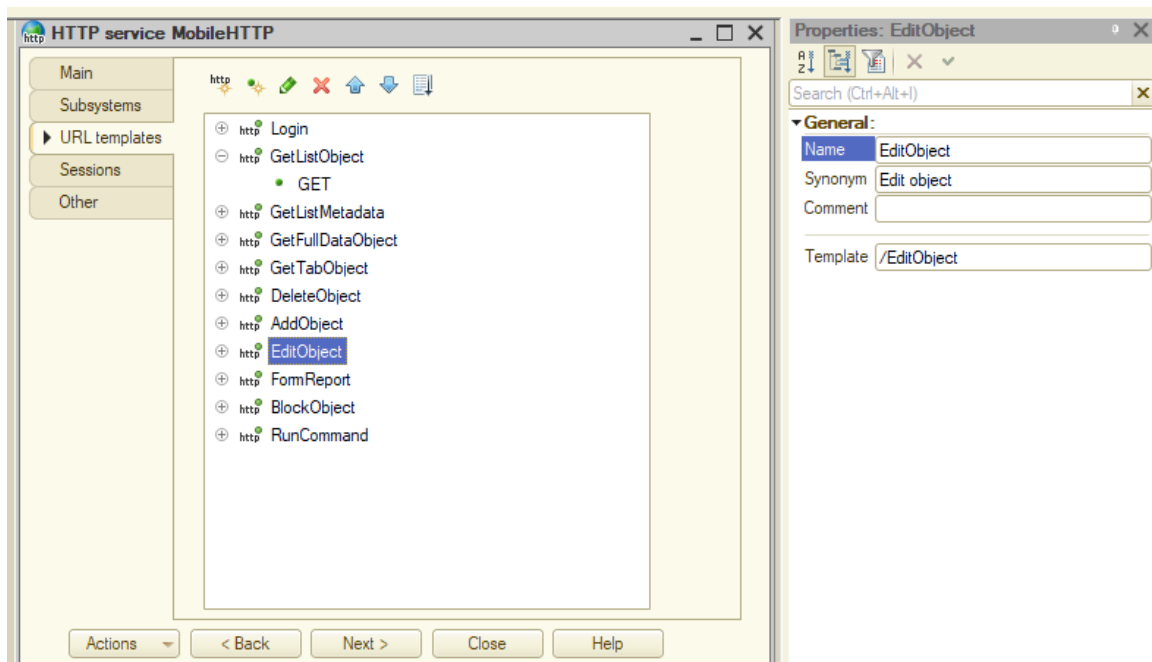


Рисунок 3.19 – Налаштування методу EditObject

Система передбачає відображення звітів в мобільному додатку. Вони будуть формуватися згідно налаштувань які обрав користувач в якості фільтру. Система здійснюватиме формування звіту і відобразить його в PDF форматі. Цей формат є універсальний і працює на більшості пристроїв. Для формування цієї дії буде додано метод HTTP сервісу FormReport. Скріншот його налаштувань подано на рисунку 3.20.

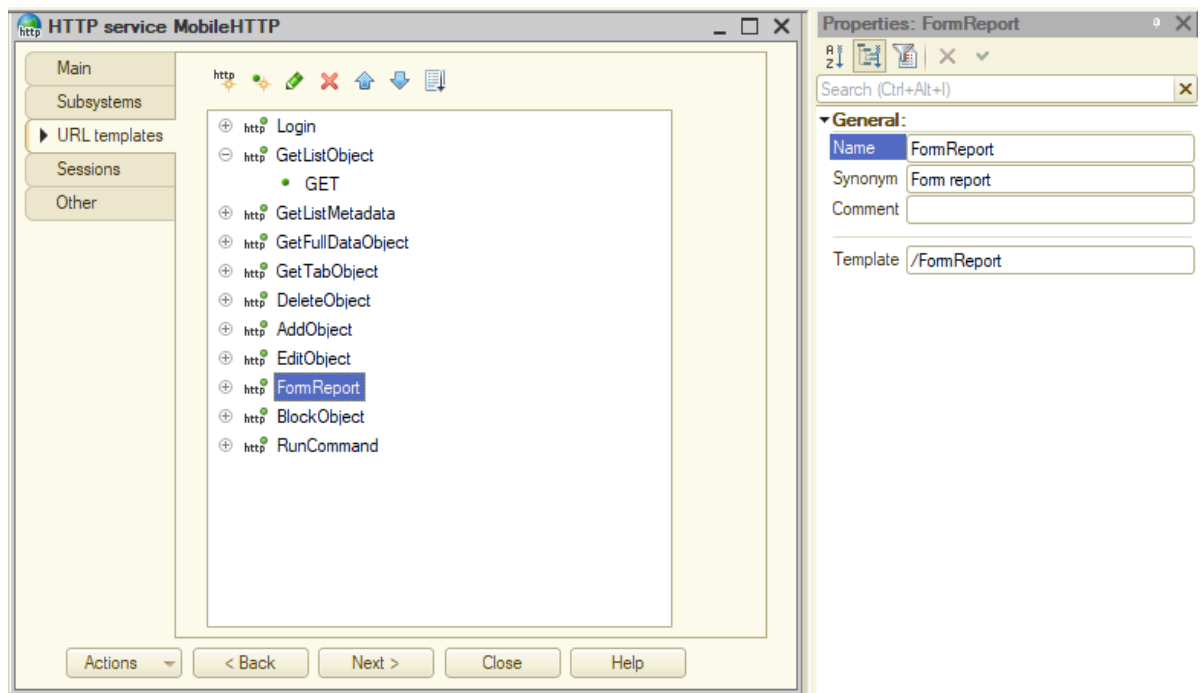


Рисунок 3.20 – Налаштування методу FormReport

Система передбачає блокування об'єктів в мобільному додатку, щоб інші користувачі не могли його редагувати. Система здійснюватиме запис у реєстр BlockedObjectsUsers, як тільки користувач його відкриє у режимі редагування. Для формування цієї дії буде додано метод HTTP сервісу BlockObject. Скріншот його налаштувань подано на рисунку 3.21.

В системі також передбачено виконання команд з певними функціями які визначить користувач на платформі VAF. Така можливість дозволить здійснювати будь які дії з об'єктами. Для цього в HTTP сервісі буде додано метод RunCommand. Зображення його налаштувань подано на рисунку 3.22.

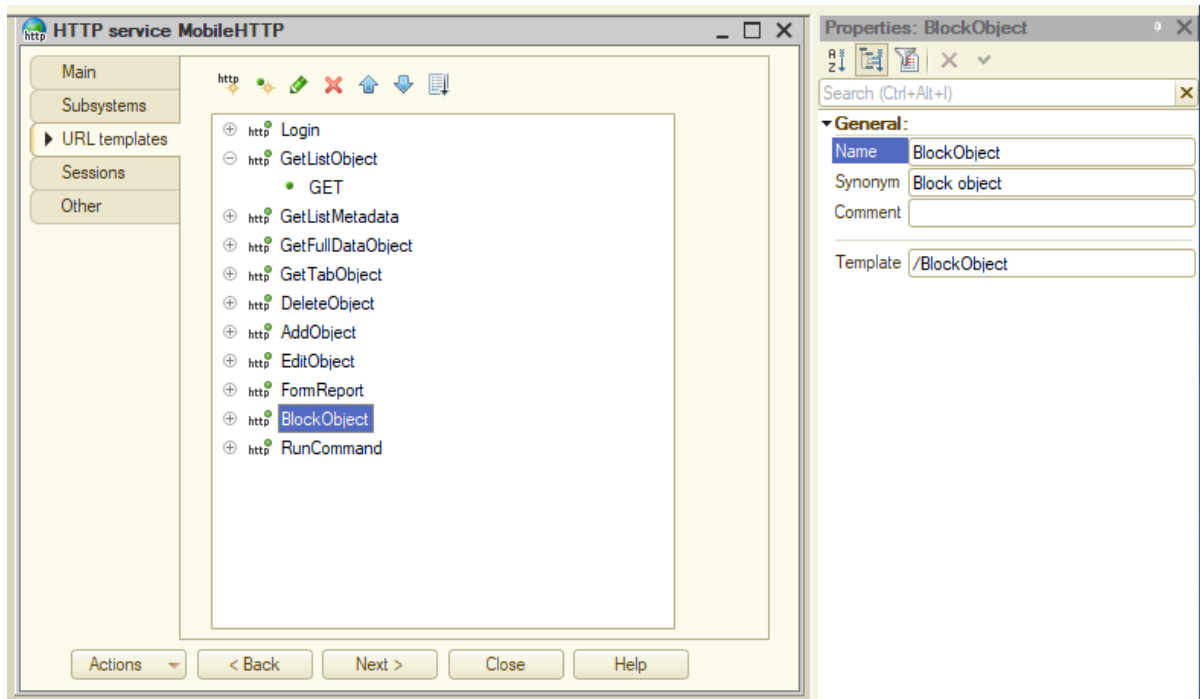


Рисунок 3.21 – Налаштування методу блокування BlockObject

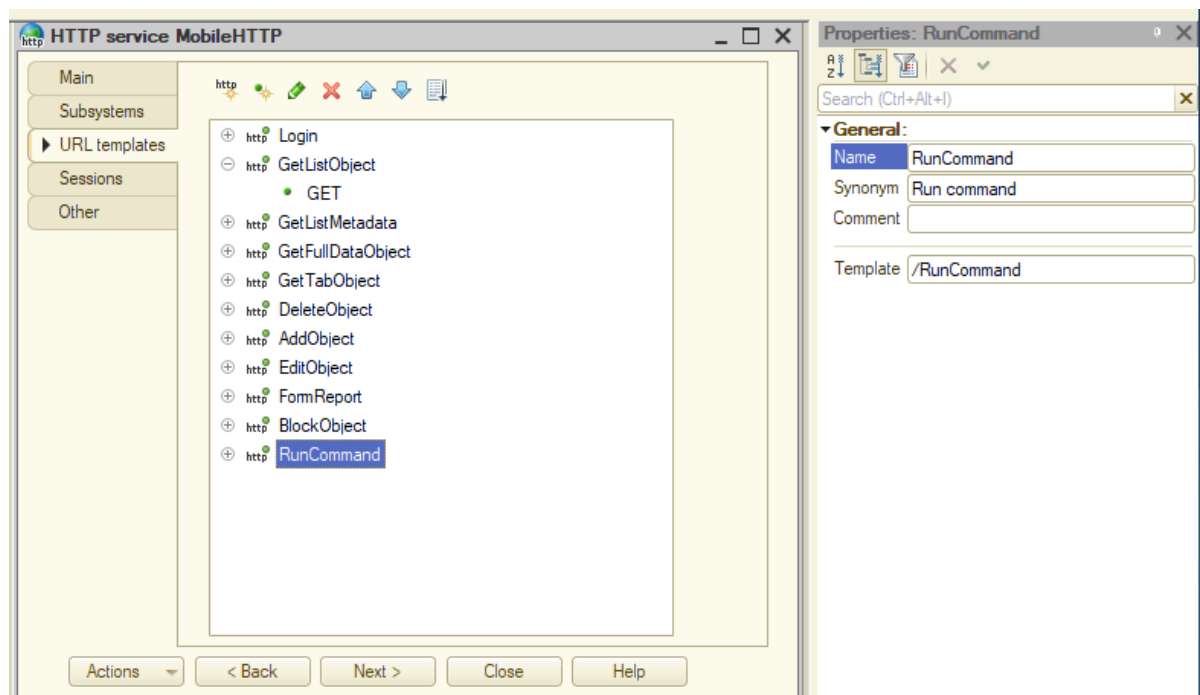


Рисунок 3.22 – Налаштування методу RunCommand

3.3 Висновок до розділу 3

Для підвищення продуктивності системи та ефективності роботи користувачів, було реалізовано новий метод опису метаданих, який дозволяє

полегшити та прискорити роботу з великими обсягами даних в системі. Для цього було створено HTTP-сервіс в платформі VAF, який отримав назву MobileHTTP.

Для налаштування HTTP-сервісу в платформі VAF використовується ієрархічний довідник, який містить загальні налаштування та налаштування, специфічні для конкретних об'єктів. Це забезпечує гнучкість у конфігурації системи.

Для забезпечення цілісності даних та координації паралельних операцій було створено реєстр відомостей BlockedObjectsUsers, який відслідковує блокування об'єктів користувачами.

Структура роботи HTTP-сервісу в поєднанні з платформою VAF дозволяє ефективно обробляти запити мобільного застосунку, використовуючи раніше створені елементи.

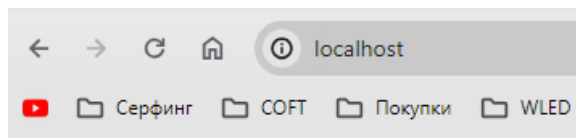
Для забезпечення безпечної авторизації було використано вбудовану функцію "Безпечна авторизація", яка захищає передачу паролів користувачів.

4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ТА ТЕСТУВАННЯ СИСТЕМИ

4.1 Тестування запуск серверної частини

Для успішного запуску системи необхідно запуснути його серверу частину. Для її запуску необхідно в конфігураторі виконати публікацію на веб-сервері Apache[35]. Сам процес складається з наступних кроків:

1. Встановити та налаштувати веб-сервер Apache сервері, для цього потрібно скористатися інструкцією на офіційному сайті[21].
2. Переконайтеся, що Apache коректно працює та може обслуговувати HTTP-запити.



It works!

Рисунок 4.1 – Перевірка роботи сервера Apache

3. Налаштувати Apache для публікації. Для цього потрібно перейти до конфігурації веб-сервера, яка знаходиться у файлі conf.
4. Увімкніть опцію публікації веб-сервера та налаштувати необхідні параметри, такі як IP-адреса, порт тощо.
5. У платформі ВАФ виберіть пункт "Publish to server" та оберіть "Публікувати на веб-сервері Apache".

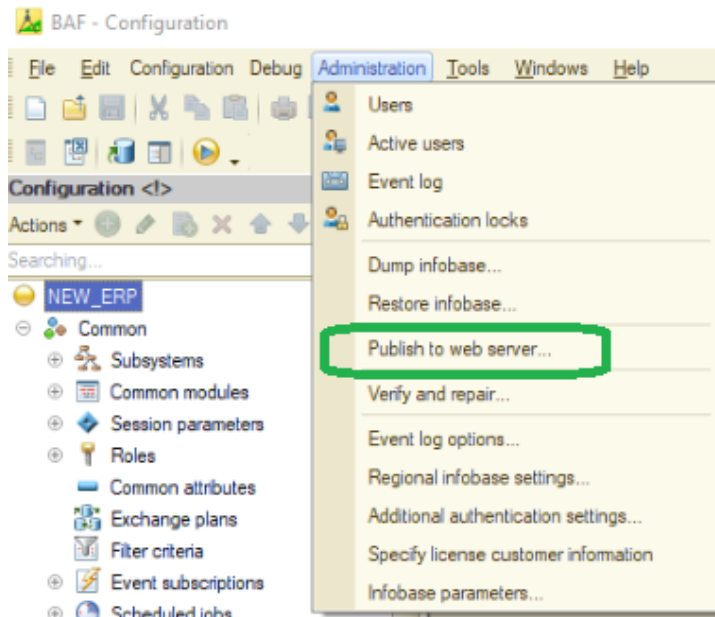


Рисунок 4.2 – Налаштування в платформі BAF

6. Вкажіть шлях до кореневої директорії Apache, куди будуть скопійовані файли налаштувань.

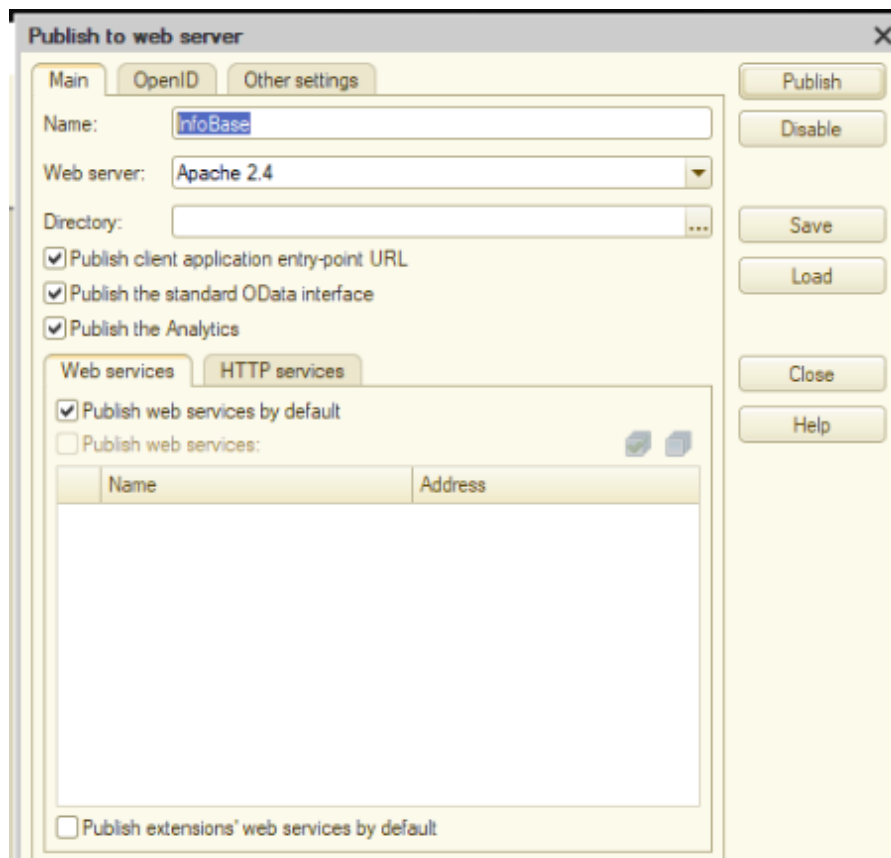


Рисунок 4.3 – Налаштування публікації в платформі BAF

Натиснути "Publish" для запуску процесу публікації.

4.2 Тестування та налаштування платформи ВАФ

Для здійснення налаштування мобільного додатку в режимі користувача потрібно здійснити авторизацію. Далі перейти в підсистему адміністрування і ній обрати довідник «Налаштування обмінів mobile». Та натиснути кнопку «Create group». Скріншоти створення та заповнення елемента групи довідника подано на рисунку 4.4-4.5.

☆ Налаштування обмінів mobile (create... [link](#) [refresh](#) [close](#))

Save and close Save More actions ▾

Code:

Description:

Parent: ▾ [copy](#)

Синоним:

Зображення: ▾

Рисунок 4.4 – Налаштування в платформі ВАФ

☆ Directories (Налаштування обмінів m... [link](#) [refresh](#) [close](#))

Save and close Save More actions ▾

Code:

Description:

Parent: ▾ [copy](#)

Синоним:

Зображення: ▾

Рисунок 4.5 – Налаштування в платформі ВАФ

Окрім створення групи потрібно додати ще елементи довідника. Для цього необхідно натиснути кнопку «Create». Скріншоти створення та заповнення

елемента довідника подано на рисунку 4.6-4.7. Розглянуто для прикладу опис об'єкту метаданих довідники “Контрагенти”.

Рисунок 4.6 – Налаштування в платформі ВАФ

#	Ім'я	Синонім	Порядок у списку	Порядок у формі елемента
1	Телефон	Телефон	3	
2	ДеньРождения	ДР	4	
3	ФизЛицо	Фіз особа	5	
4	ИмяПредопределенныхДанных			
5	Предопределенный			
6	Ссылка			
7	ПометкаУдаления	Помітка видалення		

Рисунок 4.7 – Налаштування в платформі ВАФ

Для роботи в табличних частин потрібно додати значення в таблицю. Яке буде складатися з назви таблиці і її реквізиту. Скріншоти додавання значень для елемента довідника подано на рисунку 4.8 - 4.9.

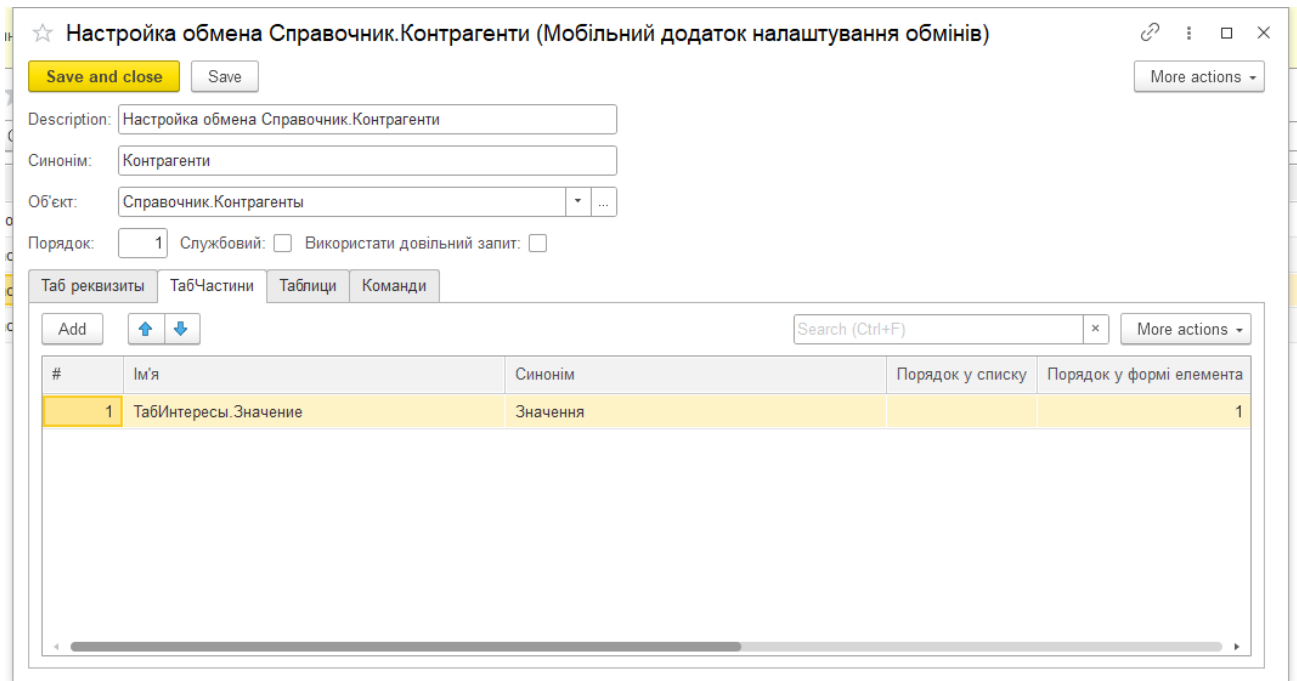


Рисунок 4.8 – Налаштування в платформі VAF

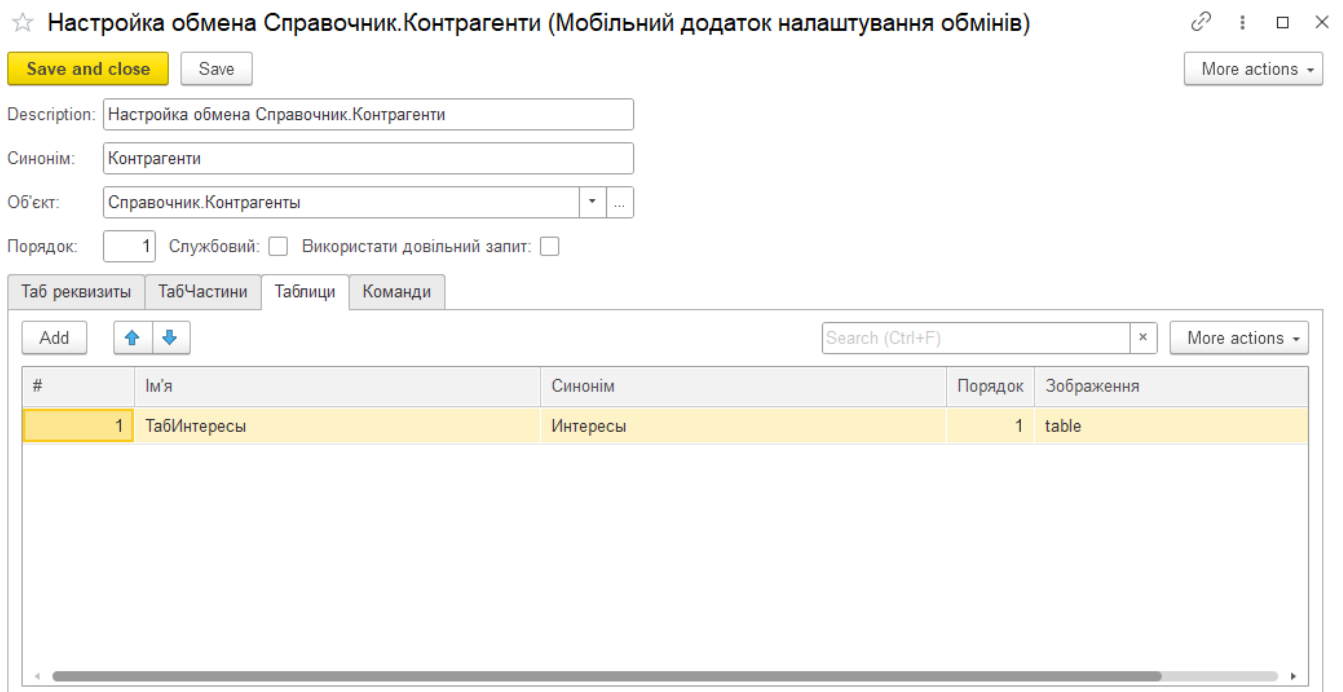


Рисунок 4.9 – Налаштування в платформі VAF таблиці

Для відображення клавiш потрібно додати список команд які буде використовувати цей об'єкт. Кожна з команд має своє призначення. Вони можуть

бути List, Element, NewElement. Скріншоти з їх додавання для елемента довідника подано на рисунку 4.10.

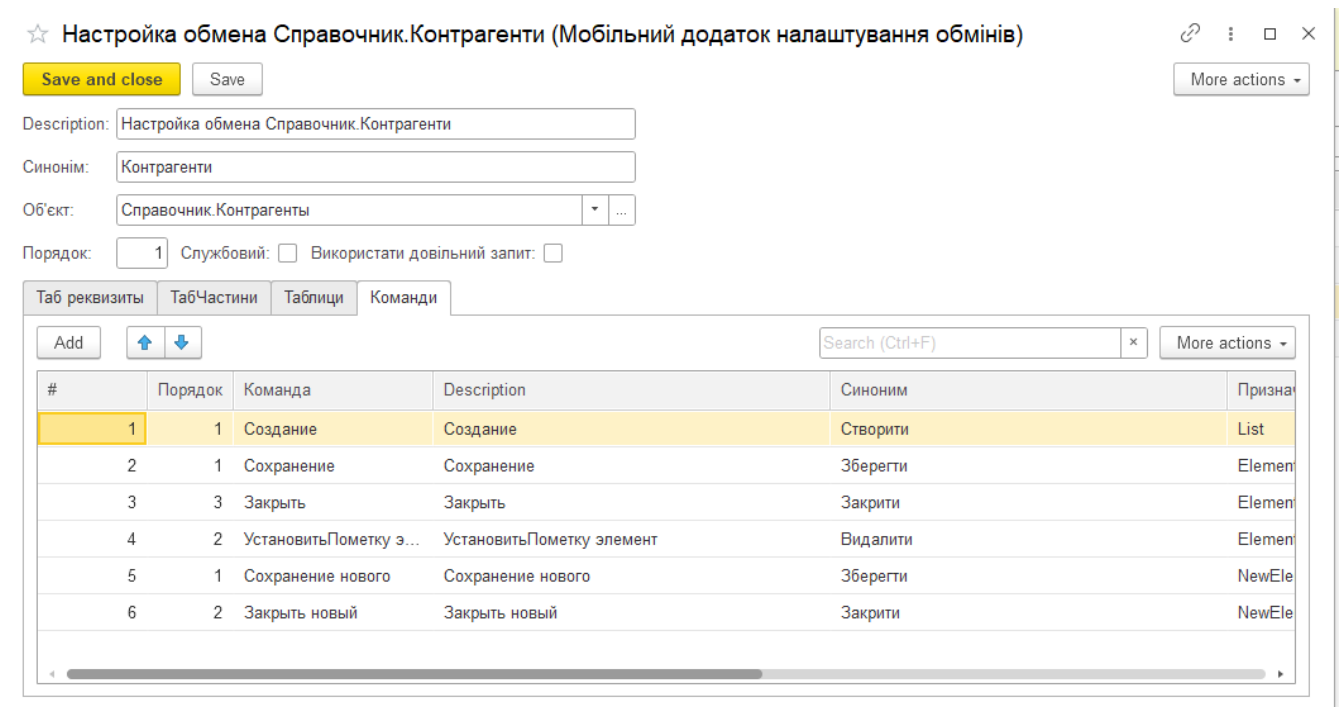


Рисунок 4.10 – Налаштування в платформі VAF команди

Розглянемо створення команд, для цього потрібно додати ще елементи довідника «Commands». Потрібно натиснути кнопку «Create». Після чого відкриється вікно нового елемента довідника команди, як показано на рисунку 4.11. Далі необхідно заповнити всі поля як це зроблено на рисунку 4.12.

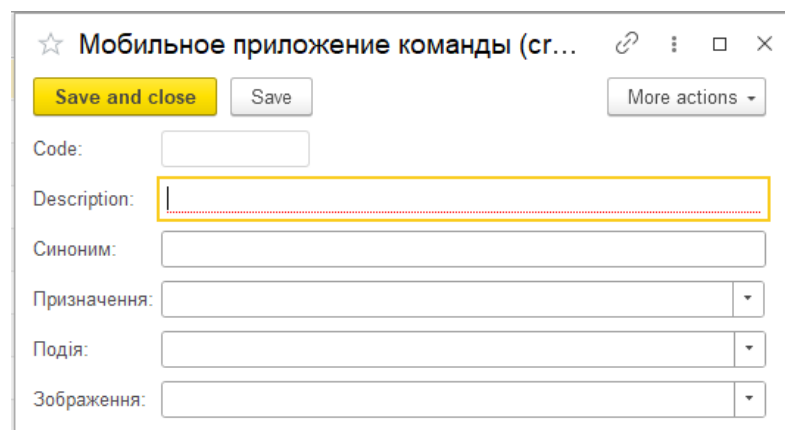


Рисунок 4.11 – Налаштування в платформі VAF

☆ Установить Пометку список (Мобильн... [🔗](#) [⋮](#) [🗑](#) [✕](#)

Save and close Save More actions ▾

Code:

Description:

Синоним:

Призначення:

Подія:

Зображення:

Рисунок 4.12 – Налаштування в платформі VAF

4.3 Сценарії роботи користувача з системою

Оскільки розроблена система працює на операційній системі Android та IOS, її можна легко та швидко встановити на смартфон, який відповідає цим мінімальним вимогам.

Для коректної роботи розробленої системи на смартфоні необхідно, щоб на пристрої була встановлена операційна система Android версії не нижче 5.0, а також мали бути доступні сервіси Google Play. Крім того, на внутрішньому сховищі пристрою має бути вільне місце обсягом не менше 50 МБ.

Для встановлення додатку на смартфон потрібно завантажити APK-файл та виконати його. Однак, для успішного встановлення, в налаштуваннях пристрою необхідно дозволити встановлення з невідомих джерел.

Розроблений додаток також потребує надання користувачем наступних дозволів:

1. дозвіл на читання та запис на зовнішньому сховищі;
2. доступ до інтернету;
3. доступ до камери.

Інтерфейс мобільного додатку складається з декількох робочих вікон, кожне з яких призначене для керування відповідною частиною CRM додатку.

На рисунку 4.13 зображено екран авторизації, який є першим, що побачить користувач при запуску додатку. Верхню частину екрану займає панель з кнопкою налаштувань, а основними кольорами інтерфейсу є темно-синій, синій та білий.

Для авторизації користувачів використовується їхнє ім'я користувача в платформі ВАР. Обов'язковою вимогою до логіна є наявність одного слова.

Пароль не має обмежень щодо свого складу, однак повинен містити не менше шести символів. Такі вимоги до довжини пароля є стандартною практикою для багатьох додатків.

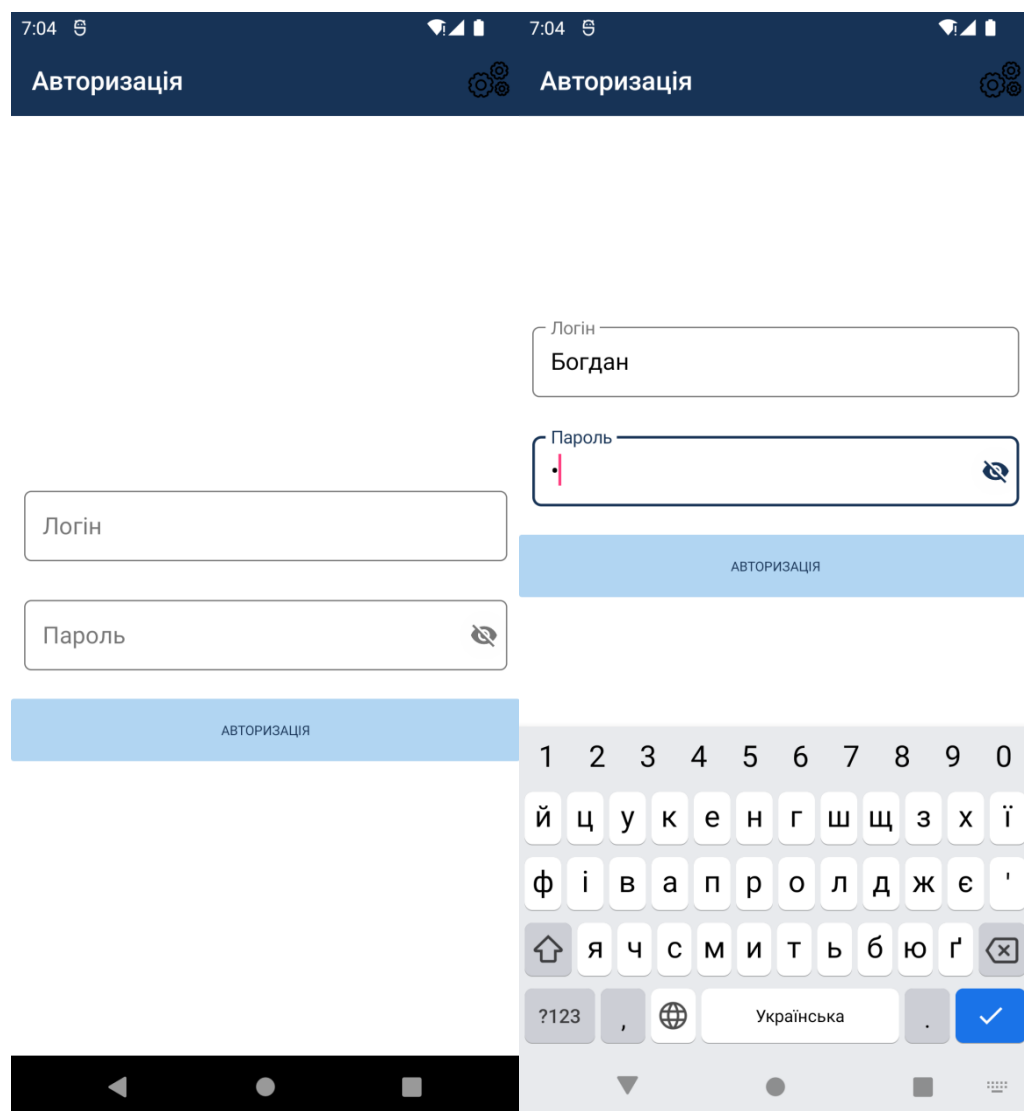


Рисунок 4.13 – Авторизація у мобільному додатку

У випадку якщо користувач введе некоректні дані в поля логіну чи пароля, ці поля будуть підсвічені з відповідним повідомленням про помилку.

Після успішної авторизації користувач потрапляє до свого списку об'єктів в мобільному додатку. На рисунку 4.14 список об'єктів які зображаться користувачу відповідно його налаштувань і прав доступу RLS. Список всіх налаштувань подано на рисунку 4.15.

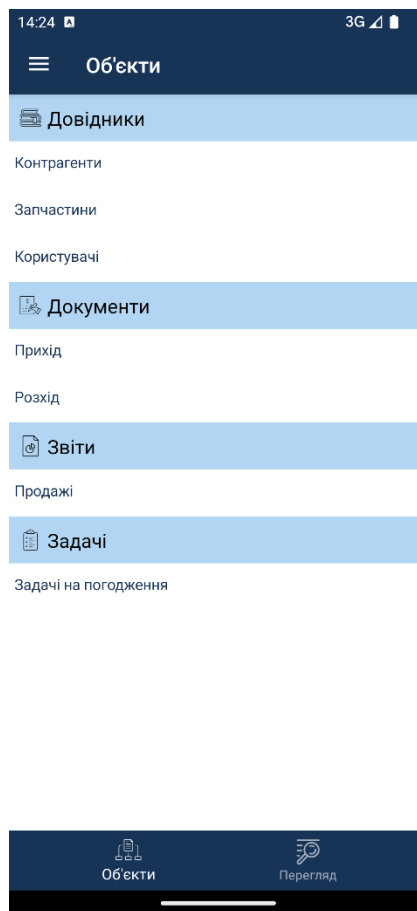


Рисунок 4.14 – Список об'єктів у мобільному додатку

☰ Довідники	00000002		Довідники	directories
▀ Настройка обмена Справочник.ЗЧ	00000005	Справочник.ЗЧ	2	Запчастини
▀ Настройка обмена Справочник.Контрагенти	00000004	Справочник.Контрагенты	1	Контрагенти
▀ Настройка обмена Справочник.Пользователи	00000003	Справочник.Пользователи	3	Користувачі
☰ Документи	00000006		Документи	document
▀ Настройка обмена Документ.Поступление	00000007	Документ.Поступление	1	Прихід
▀ Настройка обмена Документ.Расход	00000008	Документ.Расход	2	Розхід
☰ Задачі	00000011		Задачі	task
▀ Настройка обмена Задача.ЗадачаСогласован...	00000012	Задача.ЗадачаСогласование	1	Задачі на погодження
☰ Звіти	00000009		Звіти	report
▀ Настройка обмена Отчет.Продажи	00000010	Отчет.Продажи	1	Продажі

Рисунок 4.15 – Список налаштувань для мобільного додатку

На даному екрані зображена всі налаштування. На першому рівні відображають групи елементів такі як: Довідники, Документи, Звіти та Задачі. На другому рівні вже їх підлеглі елементи, вони також зазначені на рисунку 4.15.

Далі користувач обирає об'єкт з яким хоче працювати. Наприклад “Контрагенти”. На наступному екрані буде зображено список об'єктів з типом контрагент, які були додані в платформи ВАФ.

Всі колонки відряджаються по порядку згідно налаштувань які було вказано попередньо в довіднику “Settings Exchange Mobile” зображення подано на рисунку 4.15.



Код	Імя	Телефон
ДР	Фіз особа	
01.01.1900	Ні	
000000121	Петро	3889887854
02.01.2098	Ні	
000000122	Петров	38898878548
02.01.2021	Так	
000000123	Петров	38898878548
01.01.1900	Ні	
000000124	пре Ні	12558
000000125	пре Ні	12558
000000126	Саша	380657458
01.01.1951	Так	
000000127	Саша	380657458
01.01.1951	Так	
000000128	Юра	380652874523
01.01.1906	Так	
000000129	Юра	380652874523
01.01.1906	Так	
000000130	Юрій	380695478653
01.01.2002	Так	
000000131	Юрій	380695478653
01.01.2002	Так	

Рисунок 4.17 – Список об'єктів «Контрагенти» у мобільному додатку

З екрану списку об'єктів також можна перейти на екран детального перегляду об'єкту (рисунок 4.19) або його створення якщо натиснути кнопку “Створити”(рисунок 4.18) . На кожній з цих сторінок відображаються лише ті команди які були вказані в довіднику налаштувань.

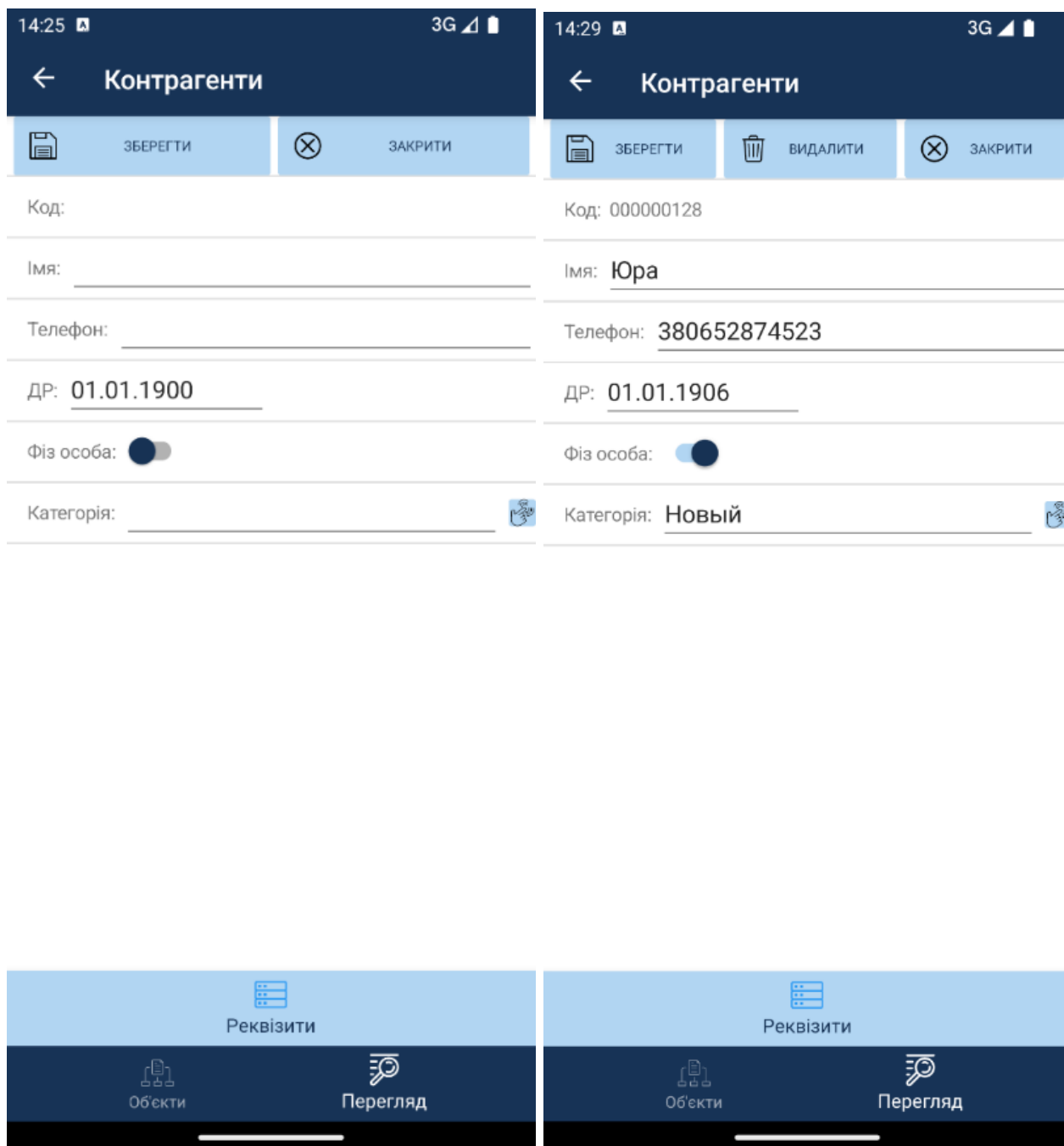


Рисунок 4.18-4.19 – Створення нового об'єкта «Контрагент» для мобільного додатку

В застосунку передбачено варіант коли користувачі можуть блокувати об'єкт. Про це буде здійснюватися запис в реєстр “BlockedObjectsUsers”. Що цей об'єкт було заблоковано. Тому він буде відображатися в режимі лише для читання. Приклад такого випадку подано на рисунку 4.20.

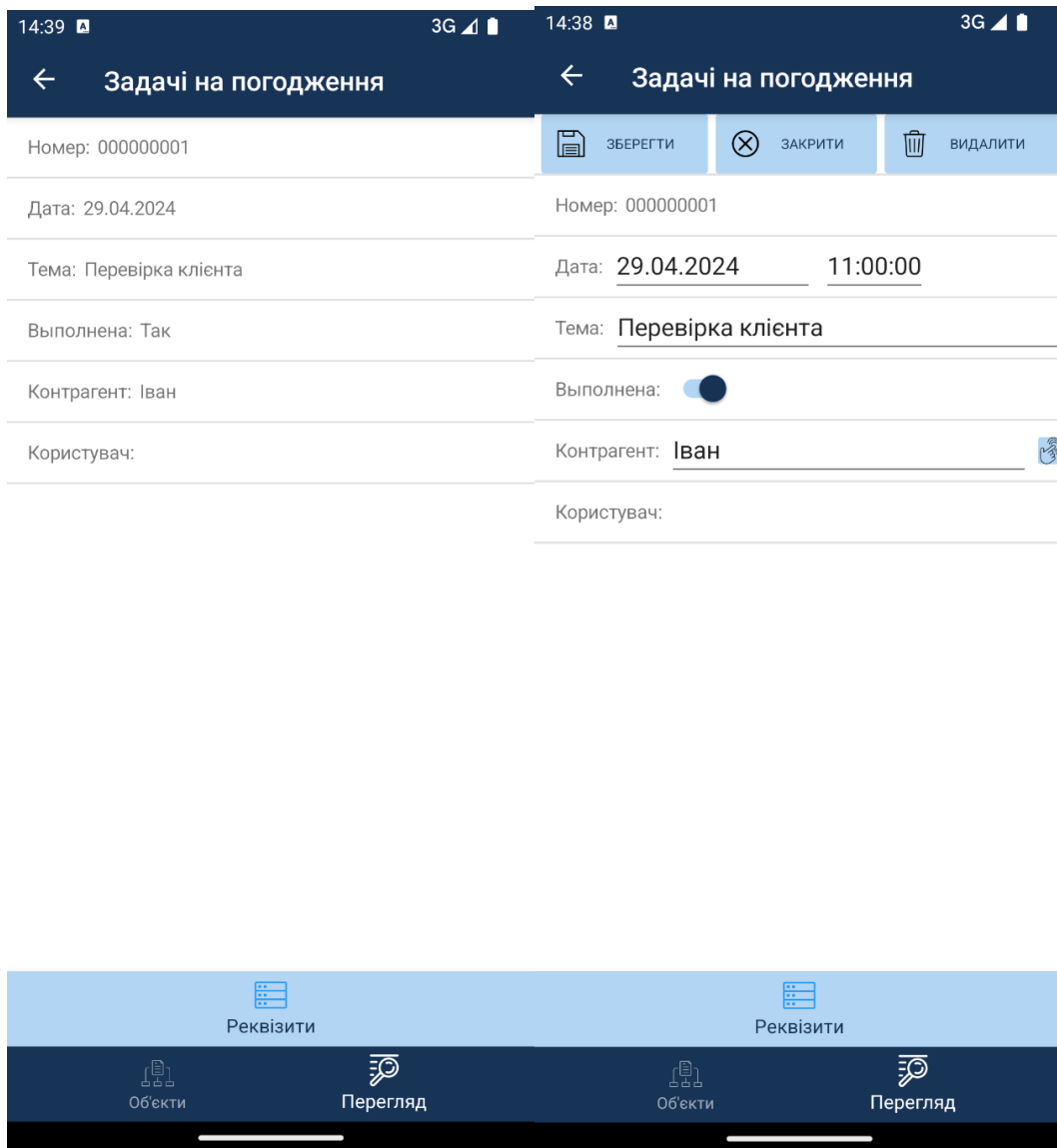


Рисунок 4.20 – Доступний для редагування та заблокований об'єкт у мобільного додатку

Для формування звітів у форматі PDF на мобільному застосунку. Система буде генерувати звіти у форматі PDF на основі даних, зібраних в BAF. PDF звіти повинні матимуть чітку структуру, логічне оформлення та бути зручними для читання.

Повинна бути можливість налаштування шаблонів PDF звітів, щоб забезпечити узгодженість оформлення і можливість встановлювати параметри вибору.

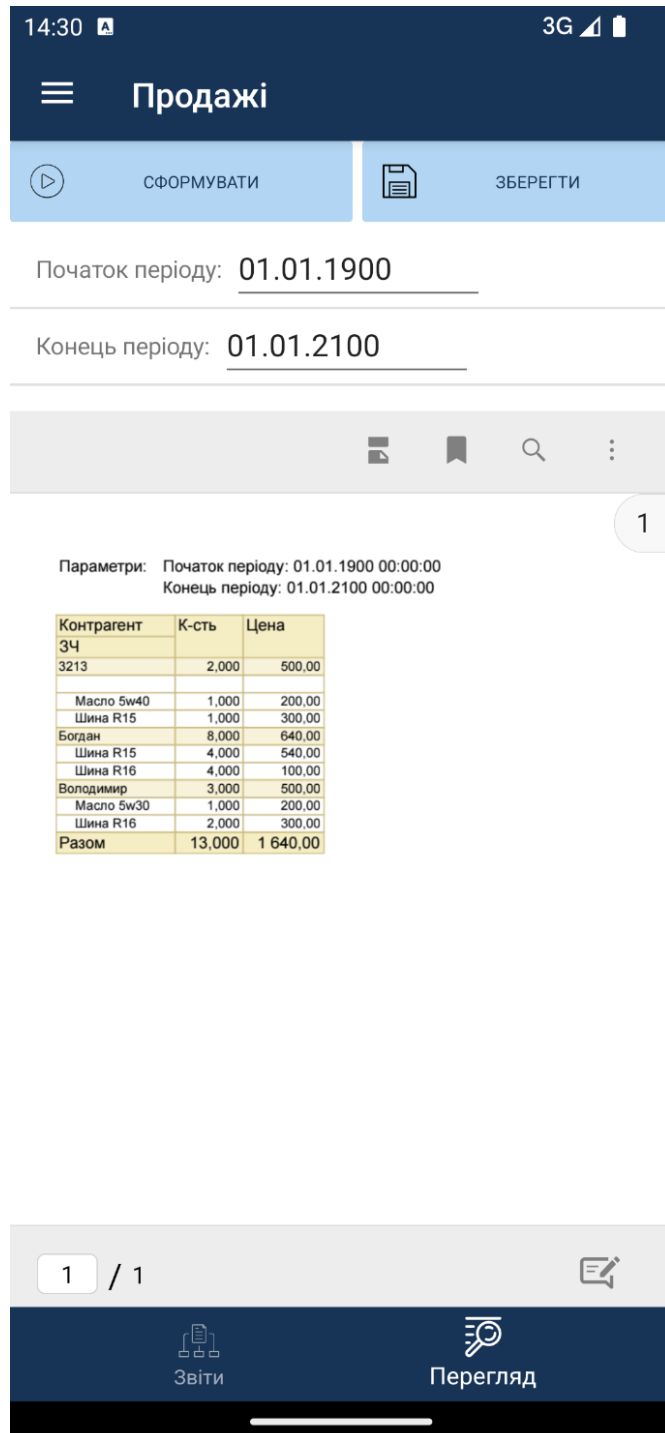


Рисунок 4.21 – Сформований звіт у мобільному додатку

4.4 Оцінка швидкодії запропонованого методу

Для оцінки та порівняння ефективності розробленого методу отримання метаданих та OData, слід використовувати різні набори даних та сценарії. Це дозволить точно оцінити ефективність методів та зробити обґрунтовані висновки. Для ефективного порівняння продуктивності потрібно використати програмне забезпечення Postmen, яке дозволяє вимірювати час виконання наступних операцій: додавання елементів в масив, видалення елементів з масиву, редагування елементів масиву. Комбінація цих операцій дозволить детально дослідити переваги та недоліки методу в різних ситуаціях.

Варто зазначити, що звичайній простій операції над масивом з невеликою кількістю елементів виконуються дуже швидко. Тому потрібно проводити тестування на масивах з більшою кількістю елементів. Перед початком тестування, стан кожного методу буде відповідати пустому масиву.

Для коректного тестування продуктивності методів, рекомендується проводити сценарії роботи декілька разів. Для отримання показника продуктивності буде використано середнє арифметичне з трьох випробувань. Це дозволить уникнути можливих похибок та сторонніх ефектів під час роботи програмного забезпечення.

Для дослідження будуть використані наступні сценарії тестування продуктивності:

- читання 100, 500, 1000, 10000 елементів;
- видалення об'єктів значень 1, 10, 100, 500, 1000 елементів;
- зміна значень а 1, 10, 100, 500, 1000;
- комбінація додавання, видалення та редагування елементів в довільному порядку 1, 10, 100, 500, 1000.

Отже, проаналізувавши сценарії тестування продуктивності методів, можна приступити до проведення тестування з використанням розробленого програмного забезпечення та інструментів Postmen. Результати роботи представлені у вигляді таблиць і діаграм для наочного порівняння. На основі результатів буде зроблено

висновки про доцільність використання розробленого методу в різних ситуаціях. Ці результати також були використані при розробці мобільного додатку.

Таблиця 4.1 Результати читання даних з платформи ВAF мс

Кількість елементів	Час проведення операції додавання елементів використовуючи метод, мс	
	OData	Розроблений метод
100	322	230
500	12045	8424
1000	20457	14578
10000	198624	135476

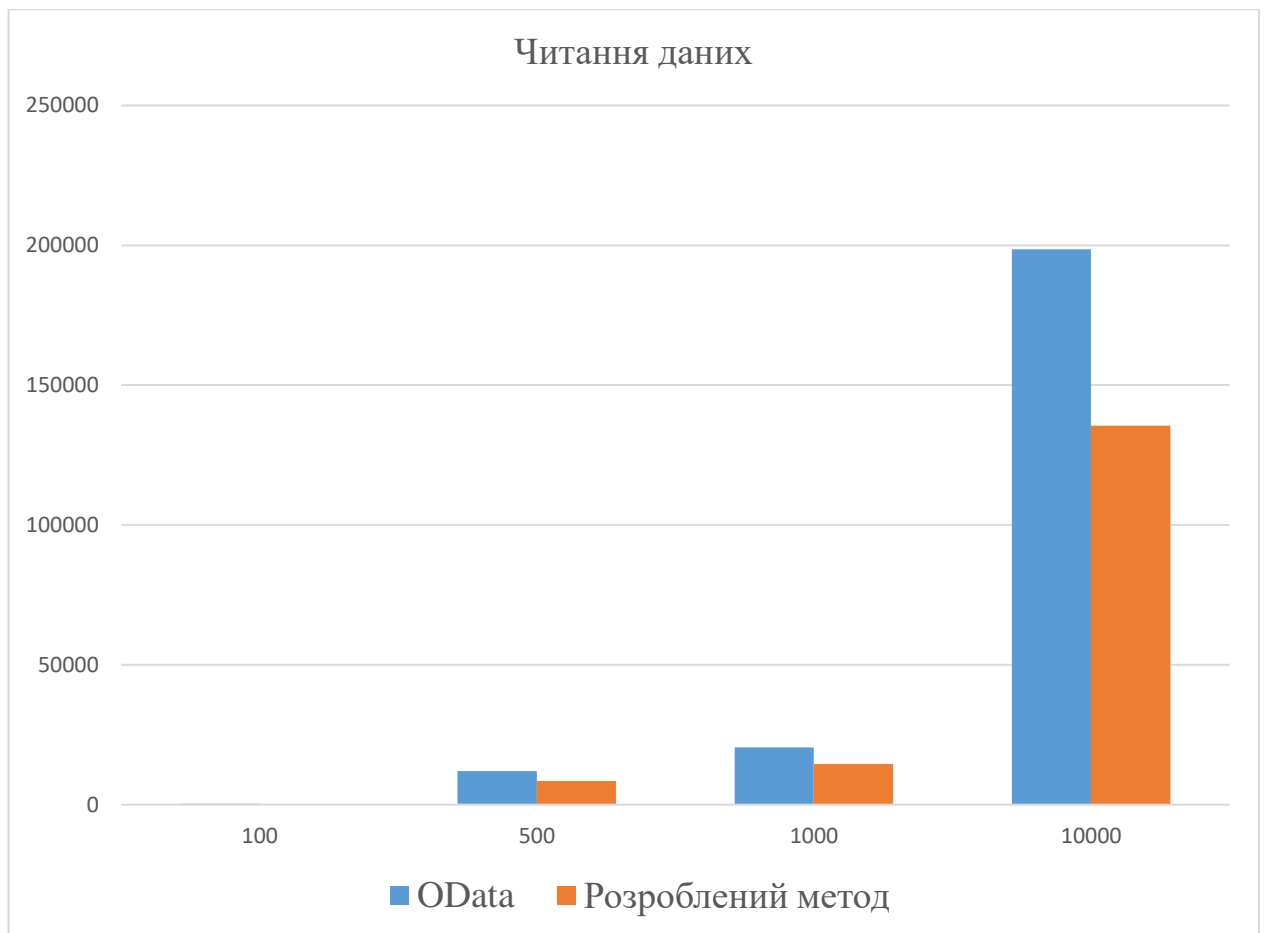


Рисунок 4.21 – Діаграма порівняння операції читання даних з платформи ВAF

Отримавши діаграми (рисунок 4.21), при читанні даних з платформи ВAF найкращий результат показав розроблений метод. Якщо порівняти результати при

більшій кількості елементів на 10000 значень розроблений метод має значну перевагу в часі на 44%.

Таблиця 4.2 Результати видалення даних з платформи ВАР мс

Кількість елементів	Час проведення операції додавання елементів використовуючи метод, мс	
	OData	Розроблений метод
1	16	16
10	151	152
100	1426	1419
500	5821	5832
1000	9443	9433

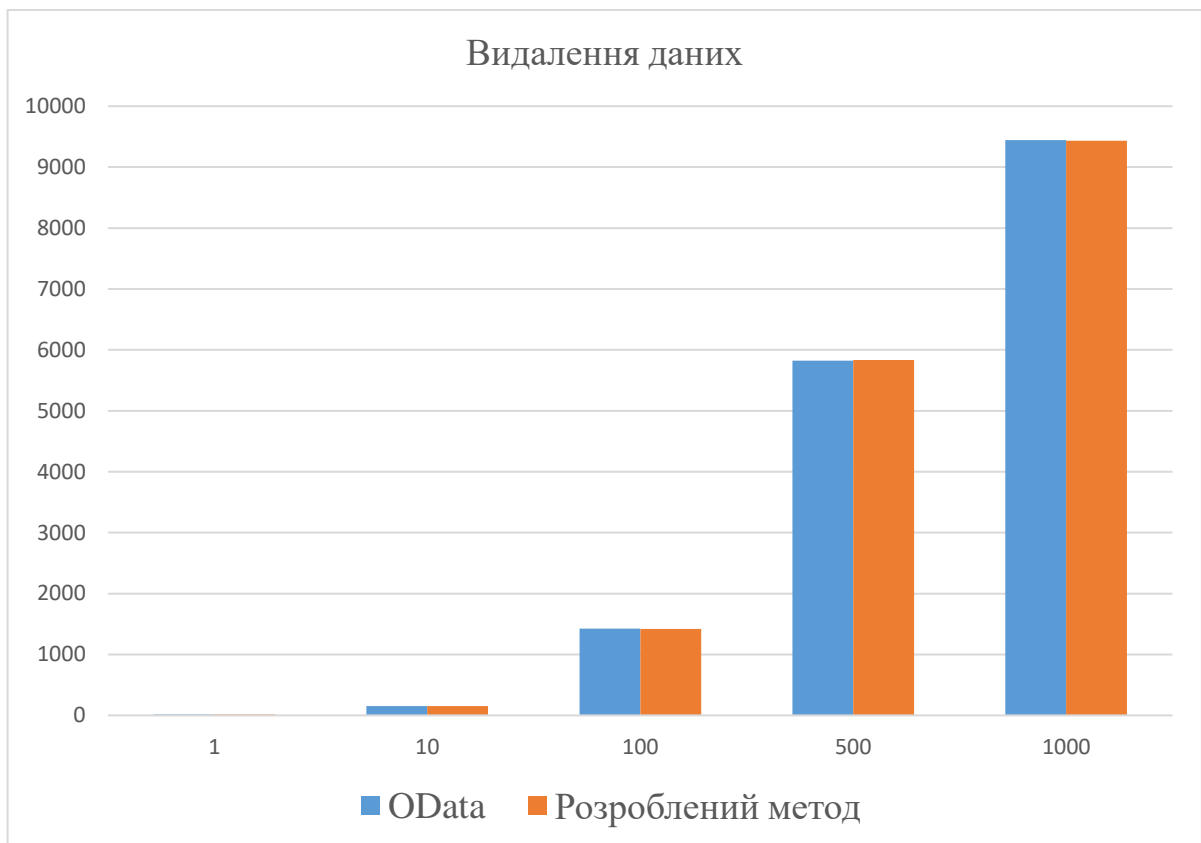


Рисунок 4.22 – Діаграма порівняння операції видалення даних з платформи ВАР

Як можна побачити з діаграми (рисунок 4.22), при видаленні даних з платформи ВАФ час виконання маже однаковий у двох випадках. Якщо порівняти результати при більшій кількості елементів на 10000 значень вони теж однакові.

Таблиця 4.3 Результати зміни даних з платформи ВАФ мс

Кількість елементів	Час проведення операції додавання елементів використовуючи метод, мс	
	OData	Розроблений метод
10	161	142
100	1322	1212
500	5448	4845
1000	9024	8421

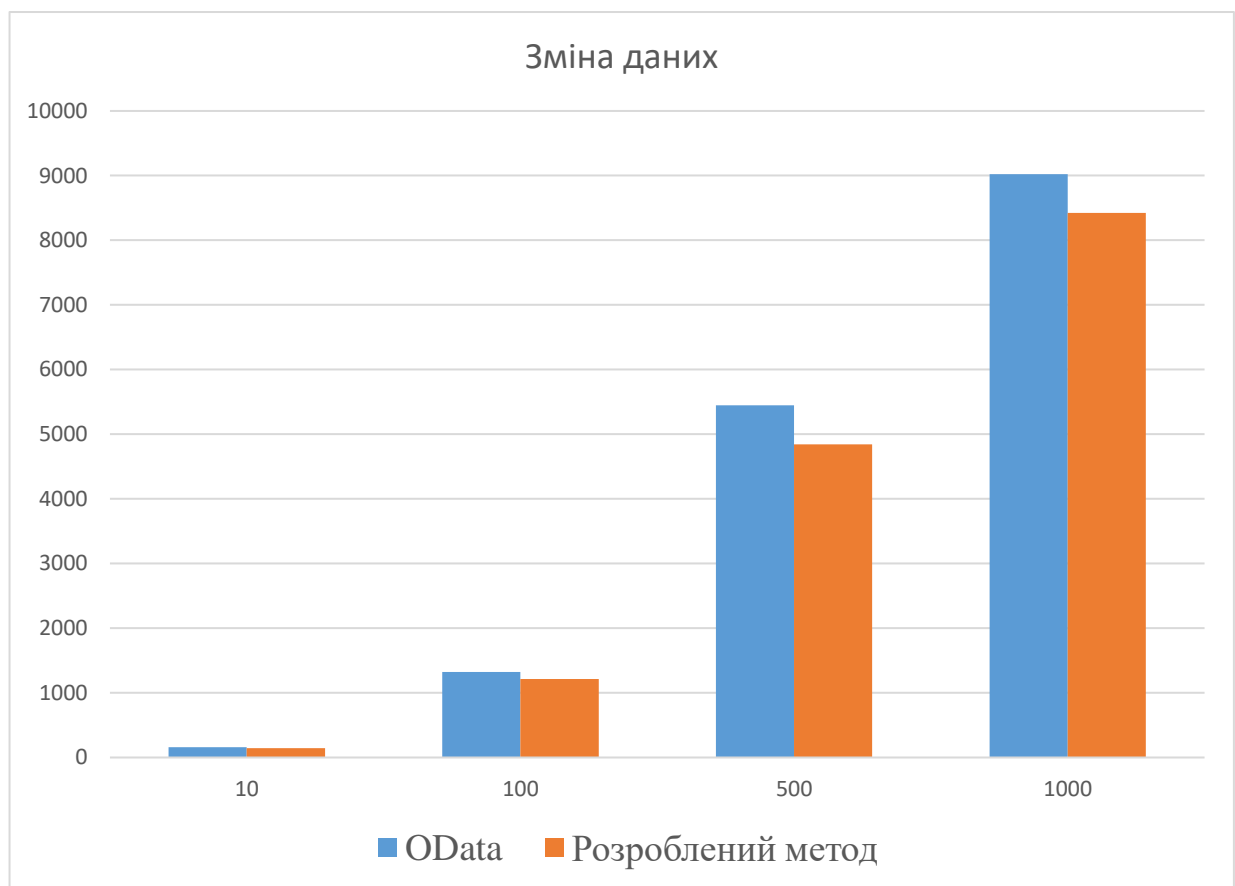


Рисунок 4.23 – Діаграма порівняння операції зміни даних з платформи ВАФ

Аналізуючи дані отримані діаграми (рисунок 4.23), при зміні даних з платформи ВАФ найкращий результат швидкодії показав розроблений метод. Якщо порівняти результати при більшій кількості елементів на 1000 значень розроблений метод має значну перевагу в часі на 7%.

Таблиця 4.4 Результати комбінації дій над даними в платформі ВАФ

Кількість елементів	Час проведення операції додавання елементів використовуючи метод, мс	
	OData	Розроблений метод
1	66	42
10	432	320
100	3214	2144
500	12747	8124
1000	21457	14578

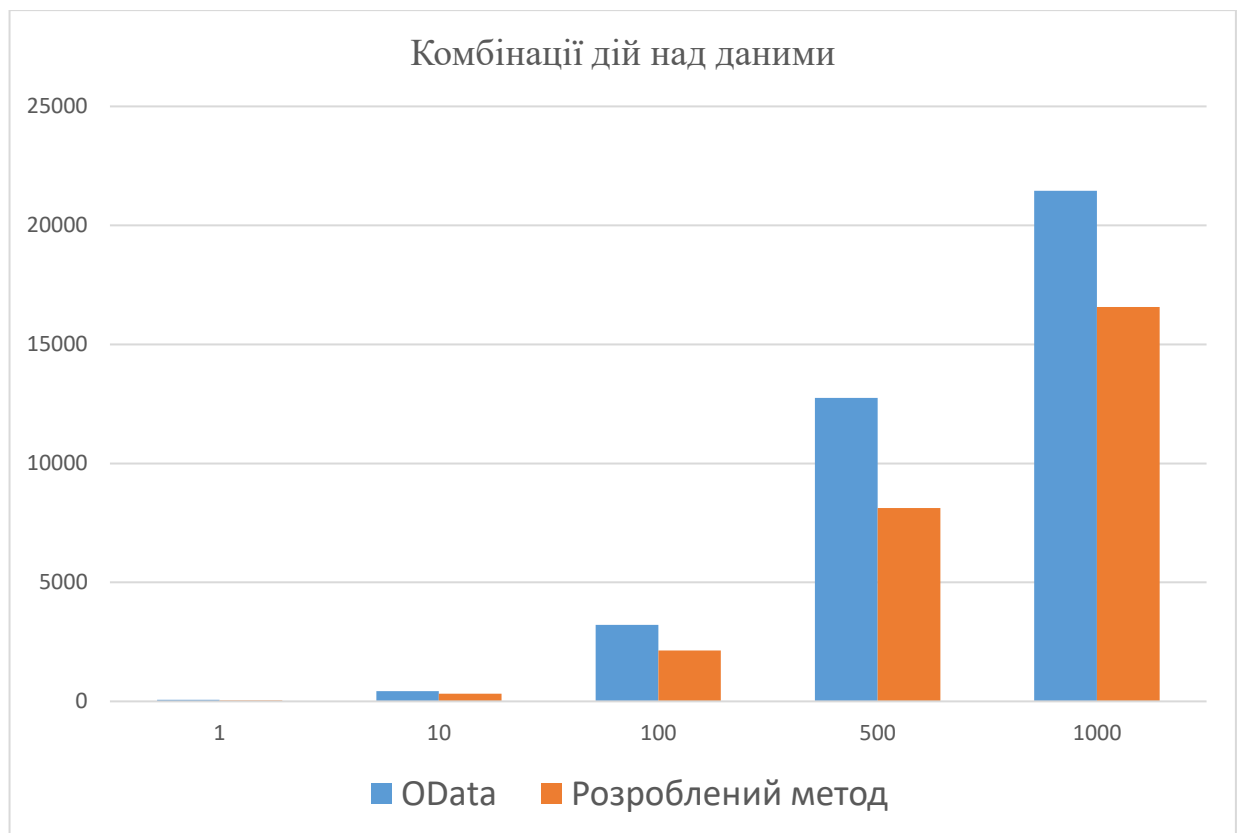


Рисунок 4.24 – Діаграма порівняння комбінації дій над даними у платформі ВАФ

Отримавши діаграму (рисунок 4.24), при комбінації дій над даними з платформи VAF найкращий результат показав розроблений метод. Якщо порівняти результати при більшій кількості елементів на 1000 значень розроблений метод має значну перевагу в часі на 47%.

4.5 Висновок до розділу 4

В розділі описано процес тестування та налаштування системи, а також ґрунтовне порівняння ефективності розробленого методу, що дозволяє зробити обґрунтовані висновки щодо доцільності його використання. Здійснено успішний запуск системи вимагає налаштування та запуску веб-сервера Apache. Процес включає встановлення та конфігурацію Apache, перевірку його коректної роботи, налаштування публікації веб-сервера в платформі VAF.

Для роботи мобільного додатку в режимі користувача необхідно виконати авторизацію та налаштувати довідник "Налаштування обмінів mobile". Створення та налаштування елементів довідників, таблиць, команд є важливими кроками для коректної роботи системи.

Мобільний додаток розроблений для операційних систем Android та iOS, має необхідні мінімальні вимоги до пристрою. Описано процес встановлення додатку, необхідні дозволи, інтерфейс та основні сценарії роботи користувача.

Проведено тестування та порівняння ефективності розробленого методу отримання метаданих та OData за такими критеріями: читання, видалення, зміна, комбінація дій. Результати тестування представлено у вигляді таблиць та діаграм, зроблено висновки про переваги розробленого методу.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі VAF.

У першому розділі проведено аналіз предметної області систем управління відносинами з клієнтами (CRM), визначено актуальні проблеми розробки та впровадження мобільних CRM-додатків, а також розглянуто роль предметної області в сучасному світі. Здійснено огляд існуючих CRM-систем та проаналізовано їх можливості.

У другому розділі для ефективного управління метаданими в сучасних інформаційних системах запропоновано використання апарату алгебри каскадів. Це дає змогу формалізувати опис структури методу опису метаданих, ввести необхідні операції над каскадами, а також розробити систему показників для всебічної оцінки ефективності функціонування методу. Застосування цього підходу сприяє підвищенню якості, доступності та цінності метаданих як важливого інформаційного ресурсу.

У третьому розділі наведено розроблений метод опису метаданих об'єктів, який передбачає використання ієрархічного довідника для централізованого управління налаштуваннями. Розглянуто структуру довідників, що зберігають інформацію про поля, таблиці, команди тощо. Запропоновано механізми забезпечення безпеки та контролю доступу, зокрема ведення журналу авторизації користувачів та блокування об'єктів.

У четвертому розділі описано процес тестування та налаштування системи, а також здійснено ґрунтовне порівняння ефективності розробленого методу опису метаданих з існуючим протоколом OData. Результати тестування, представлені у вигляді таблиць та діаграм, демонструють значне підвищення продуктивності при використанні розробленого методу, особливо при роботі з великими обсягами даних.

Набула подальшого розвитку інформаційна технологія інтеграції мобільних додатків з CRM-системами на платформі BAF з використанням мобільного додатку. Запропоновані рішення дозволяють підвищити ефективність управління взаємовідносинами з клієнтами, оптимізувати бізнес-процеси та забезпечити гнучкість у конфігурації системи.

Впровадження результатів роботи дозволили підвищити продуктивність роботи користувачів мобільної CRM-системи на платформі BAF, підвищити рівень безпеки даних та прозорість управління доступом застосунку.

За темою дипломної роботи опубліковані тези у матеріалах наукової конференції «СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ - 2024» (18-19 квітня 2024 р.) [37].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Android Profiler. Android Developers. 2019. URL: <https://developer.android.com/studio/profile/android-profiler> (дата звернення: 01.12.2023).
2. Посібник із програмування для Xamarin Forms. 2021. URL: <https://metanit.com/sharp/xamarin/> (дата звернення: 03.02.2024).
3. ARCore. Google Developers. 2019. URL: <https://developers.google.com/ar> (дата звернення: 05.02.2024).
4. ARKit. Apple. 2019. URL: <https://developer.apple.com/augmentedreality/> (дата звернення: 02.02.2024).
5. Augmented Reality. Netguru. 2019. URL: <https://www.netguru.com/blog/augmented-reality-mobile-android> (дата звернення: 11.01.2024).
6. DCCP. IETF. 2019. URL: <https://tools.ietf.org/html/rfc6773> (дата звернення: 12.04.2024).
7. Distributed Application Architecture. Sun Microsystems. 2019. URL: <https://web.archive.org/web/20110403121920/http://java.sun.com/developer/Books/jdbc/ch02.pdf> (дата звернення: 15.04.2024).
8. GPS: Global Positioning System (or Navstar Global Positioning System) Wide Area Augmentation System (WAAS) Performance Standard, Section B.3, Abbreviations and Acronyms. Archived April 27, 2017, at the Wayback Machine. URL: <http://java.sun.com/developer/Books/jdbc/ch02.pdf> (дата звернення: 18.04.2024).
9. Ontotext. Metadata Fundamentals. URL: <https://www.ontotext.com/knowledgehub/fundamentals/metadata-fundamental/> (дата звернення: 20.02.2024).
10. OpenDataSoft. What is Metadata and Why is it Important for Data Ontotext. 2021. URL: <https://www.opendatasoft.com/en/blog/what-is-metadata-and-why-is-it-important-data/> (дата звернення: 22.03.2024).

11. Fowler, M. Metadata. IEEE Software. 2015. Т. 22, № 1. С. 10-12. URL: <https://martinfowler.com/ieeeSoftware/metadata.pdf> (дата звернення: 14.03.2024).
12. HTTP. IETF. 2019. URL: <https://tools.ietf.org/html/rfc7231> (дата звернення: 26.01.2024).
13. Microsoft HoloLens. Microsoft. 2019. URL: <https://www.microsoft.com/ru-ru/holoLens> (дата звернення: 28.01.2024).
14. Mobile Augmented Reality in 2019. Medium. 2019. URL: https://medium.com/@the_manifest/mobile-augmented-reality-in-2019-6a4f687f900 (дата звернення: 30.01.2024).
15. MPEG-TS. Web Archive. 2019. URL: <https://web.archive.org/web/20080226181029/http://www.chiariglione.org/mpeg/faq/mp2-sys/mp2-sys.htm> (дата звернення: 02.03.2024).
16. MPTCP. IETF. 2019. URL: <https://tools.ietf.org/html/rfc6824> (дата звернення: 04.03.2024).
17. QUIC. Web archive. 2019. URL: <https://web.archive.org/web/20131002201320/https://src.chromium.org/chrome/trunk/src/net/quic/> (дата звернення: 03.03.2024).
18. Що таке нефункціональні вимоги: приклади, визначення. URL: <https://visuresolutions.com/uk/blog/non-functional-requirements> (дата звернення: 08.03.2024).
19. UML для бізнес-моделювання: для чого потрібні діаграми процесів. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 10.03.2024).
20. Бази даних. Поняття ER-моделі. Поняття сутності (entity). Атрибути. Види атрибутів. URL: https://www.bestprog.net/uk/2019/01/24/the-conceptof-er-model-the-concept-of-essence-and-communication-attributes-attribute-typesua/#google_vignette (дата звернення: 12.03.2024).
21. Визначення прецеденту. Що таке сценарій прецеденту та правила його опису? URL: <https://studfile.net/preview/9404039/page:6/> (дата звернення: 14.03.2024).

22. Downloading the Apache HTTP Server. URL: <https://httpd.apache.org/download.cgi> (дата звернення: 16.03.2024).

23. Building Cross-Platform iOS/Android Apps with Xamarin, Visual Studio, and C# Part 1 by Jim Wilson. URL: <https://app.pluralsight.com/library/courses/cross-platform-ios-androidvisualstudio-csharp/table-of-contents/> (дата звернення: 18.03.2024).

24. About mono. URL: <http://www.monoproject.com/docs/aboutmono/> (дата звернення: 20.03.2024).

25. Xamarin.Android application fundamentals. URL: https://developer.xamarin.com/guides/android/application_fundamentals/ (дата звернення: 22.03.2024).

26. Xamarin.Android architecture. URL: https://developer.xamarin.com/guides/android/under_the_hood/architecture/ (дата звернення: 24.03.2024).

27. Activity lifecycle. URL: https://developer.xamarin.com/guides/android/application_fundamentals/activit (дата звернення: 26.03.2024).

28. Xamarin.iOS application fundamentals. URL: https://developer.xamarin.com/guides/ios/application_fundamentals/ (дата звернення: 28.03.2024).

29. Working with Xamarin.Forms. URL: <https://developer.xamarin.com/guides/xamarin-forms/working-with/> (дата звернення: 30.03.2024).

30. Charles Petzold. Creating Mobile Apps with Xamarin.Forms First Edition. Redmond: Microsoft Press, 2016. 1187 с. URL: <https://ru.bmstu.wiki/SourceTree#.D0.9A.D0.BE.D0.BC.D0.BC.D0.B5.D0.BD.D1.82.D0.B8.D1.80.D0.BE.D0.B2.D0.B0.D0.BD.D0.B8.D0.B5> (дата звернення: 02.02.2024).

31. ЗНАЙОМТЕСЬ POSTMAN — ДЛЯ ТЕСТУВАЛЬНИКА MUST HAVE. URL: <https://www.quality-assurance-group.com/znajomtespostman-dlya-testuvalnyka-must-have/> (дата звернення: 04.02.2024).

32. HttpClient Class. URL: <https://docs.microsoft.com/ru-ru/dotnet/api/system.net.http.httpClient?view=netframework-4.8> (дата звернення: 03.02.2024).

33. Xamarin.Forms Master-Detail Page. URL: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/appfundamentals/navigation/master-detail-page> (дата звернення: 08.02.2024).

34. Ontotext. Metadata Fundamentals. URL: <https://www.ontotext.com/knowledgehub/fundamentals/metadata-fundamental/> (дата звернення: 10.02.2024).

35. OpenDataSoft. What is Metadata and Why is it Important for Data Ontotext. 2021. URL: <https://www.opendatasoft.com/en/blog/what-is-metadata-and-why-is-it-important-data/> (дата звернення: 12.04.2024).

36. Fowler, M. Metadata. IEEE Software. 2015. Т. 22, № 1. С. 10-12. URL: <https://martinfowler.com/ieeeSoftware/metadata.pdf> (дата звернення: 14.04.2024).

37. «Стан, досягнення та перспективи інформаційних систем і технологій». Матеріали конференції. С. 223-224. URL: https://www.ontu.edu.ua/download/konfi/2024/Conference_abstract-IT-2024.pdf (дата звернення: 02.05.2024).

38. Що таке метадані? (Визначення та використання). URL: <https://uk.go-travels.com/85100-metadata-definition-and-examples-1019177-9528130>(дата звернення: 02.02.2024).

39. Дублінське ядро. URL: https://uk.wikipedia.org/wiki/%D0%94%D1%83%D0%B1%D0%BB%D1%96%D0%BD%D1%81%D1%8C%D0%BA%D0%B5_%D1%8F%D0%B4%D1%80%D0%BE

40. What is metadata? . URL: <https://www.icann.org/ru/blogs/details/iana-stewardship-transition-planning-update-volume-2-10-5-2016-ru>

41. Feld C. K., Sousa, J. P., da Silva, P. M., Dawson, T. P. Indicators for biodiversity and ecosystem services: Towards an improved framework for ecosystems assessment. *Biodiversity and Conservation*. 2014. pp. 285–299.
42. Benson D. A., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., Sayers, E. W. *GenBank nucleic Acids Research*. 2015. pp. 32–37.
43. Baxevanis A. D. The importance of databases in biological discovery. *Current Protocols in Bioinformatics*. 2015. pp. 111–116.
44. Birney E. Biological database design and implementation. *Briefings in Bioinformatics*, 2018. pp. 31–38.
45. Bogen J. Experiment and observation. In: Machamer, P., Silberstein, M. (Eds.). *The Blackwell guide to the philosophy of science*. Blackwell Publishers Ltd. 2016. pp. 128–148.
46. Bourne P. Will a biological database be different from a biological journal? *PLoS Computational*. 2017. pp. 34.
47. Burge S. W., Daub, J., Eberhardt, R., Tate, J., Barquist, L., Nawrocki, E. P., Eddy, S. R., Gardnr, P. P., Bateman, A. Rfam 11.0: 10 years of RNA families. *Nucleic Acids Research*. 2022. pp. 226–232.
48. Canali S. Evaluating evidential pluralism in epidemiology: Mechanistic evidence in exposome research. *History and Philosophy of the Life Sciences*. 2014. pp. 4.
49. Dalmaris E., Avramidou, E. V., Xanthopoulou, A., Aravanopoulos, F. A. Dataset of targeted metabolite analysis for five taxanes of hellenic. *Taxus baccata L. populations*. Data. 2020. pp.22.
50. Dietrich M. R., Ankeny, R. A., Chen, P. M. Publication trends in model organism research. *Genetics*. 2017. pp. 787–794.
51. Duggirala S. NewSQL databases and scalable in-memory analytics. *Advances in Computers*. 2018. pp. 49–76.
52. Duigou T., du Lac, M., Carbonell, P., Faulon, J. L. RetroRules: A database of reaction rules for engineering biology. *Nucleic Acids Research*. 2017. pp. 229–235.
53. Elliot, K. C., Cheruvilil, K. S., Montgomery, G. M., Soranno, P. A. Conceptions of good science in our data-rich world. *Science data*. 2016. pp. 880–889.

54. Gharajeh M. S. . A learning analytics approach for job scheduling on cloud servers. In: Peña-Ayala, A. (Ed.). *Learning analytics: Fundamentals, applications, and trends*. Springer. 2019. pp. 269–302.
55. Gharajeh M. S. Biological big data analytics. *Advances in Computers*, 109, pp. 321–355.
56. Illari P., Floridi, L. Information quality, data and philosophy. In: Floridi, L., Illari, P. (Eds.). *The philosophy of information quality*. Berlin, Springer. 2017. pp. 5–7.
57. Kashyap H., Ahmed, H. A., Hoque, N., Roy, S., Bhattacharyya, D. K. (2015). Big data analytics in bioinformatics: A machine learning perspective. *Journal of Latex Class Files*. 2014. pp. 1–10.
58. Kinjo A. R., Bekker, G. J., Suzuki, H., Tsuchiya, Y., Kawabata, T., Ikegawa, Y., Nakamura, H. Protein Data Bank Japan (PDBJ): Updated user interfaces, resource description framework, analysis tools for large structures. *Nucleic Acids Research*. 2017. pp. 282–288.
59. Leonelli, S. When humans are the exception: Cross-species databases at the interface of biological and clinical research. *Social Studies of Science*. 2019. pp. 214–236.
60. Leonelli S. Global data quality assessment and the situated nature of "best" research practices in biology. *Data Science Journal*. 2020. pp. 32.
61. Leonelli S. Scientific research and big data. In: Edward, N. Z. (Ed.). *The Stanford encyclopedia of philosophy*. Stanford University, Stanford.
62. Leonelli S., Ankeny, R. A. . Re-thinking organisms: the impact of databases on model organism biology. *Studies in History and Philosophy of Science*. 2017. pp. 29–36.
63. Leonelli S., Tempini, N. Where health and environment meet: The use of invariant parameters in big data analysis. *Synthese*. 2015. pp. 245–250.
64. Nickles T. Alien reasoning: Is a major change in scientific research underway? *Topoi*. 2017. 901–914.
65. Pejić Bach M., Bertonsel, T., Meško, M., Suša Vugec, D., Ivančić, L. Big data usage in European countries: *Cluster analysis approach*. 2014. pp. 25.

66. Pietsch W. The causal nature of modeling with big data. *Philosophy and Technology*. 2019. pp. 137–171.
67. Raj P. A detailed analysis of NoSQL and NewSQL databases for big data analytics and distributed computing. *Advances in Computers*. 2016. pp. 1–48.
68. Ratti E. Big data biology: Between eliminative inferences and exploratory experiments. *Philosophy of Science*. 2022. pp. 198–218.
69. Sarita S. Kumar, G. S., Anuradaha, N., Sanjay, K., Rajendra, N., Kishore, S. P., Kumar, P. K. Comparative modeling study of the 3-D structure of small delta antigen protein of hepatitis delta virus. *Journal of Computer Science and Systems Biology*. 2021. pp. 47.
70. Shanthi V. Ramanathan, K. Sethumadhavan R. Role of the cation- π interaction in therapeutic proteins: A comparative study with conventional stabilizing forces. *Journal of Computer Science and Systems Biology*. 2018. pp. 51–68.
71. Shavit A. Griesemer J. There and back again, or the problem of locality in biodiversity surveys. *Philosophy of Science*. 2019. pp. 273–294.
72. Silva Y. N., Dietrich, S. W., Reed, J. M., Tsosie, L. M. Integrating big data into the computing curricula. In: *SIGCSE '14: Proceedings of the 45th ACM technical symposium on computer science education*. Machinery, Ney York. 2017. pp. 139–144.
73. Sterner B., Franz, N. M. Taxonomy for humans or computers? Cognitive pragmatics for. *Big data biological Theory*. 2017. pp. 99–111.
74. El Mouden, Z.A., Jakimi, A., Hajar, M., Boutahar, M. Graph Schema Storage in SQL Object-Relational Database and NoSQL Document-Oriented Database: A Comparative Study. In *Proceedings of the International Conference Europe Middle East North Africa Information Systems and Technologies to Support Learning*. Marrakech, Morocco. 2022. pp. 176–183.
75. Rathika, V. Graph-Based Denormalization for Migrating Big Data from SQL Database to NoSQL Database. In *Proceedings of the Intelligent Communication Technologies and Virtual Mobile Networks*. Tirunelveli, India. 2019. pp. 546–556.

76. Zhu, Y., Yan, E., Song, I. The use of a graph-based system to improve bibliographic information retrieval: *System design, implementation, and evaluation*. *Journal of the Association for Information Science and Technology*. 2024. pp. 480–490.
77. Jung, M.-G., Youn, S.-A., Bae, J. Choi, Y.-L. A study on data input and output performance comparison of MongoDB and PostgreSQL in the big data environment. *In Proceedings of the 2015 8th International Conference on Database Theory and Application (DTA)*. Jeju Island, Republic of Korea. 2021. pp. 54–57.
78. Fatima, H., Wasnik, K. Comparison of SQL, NoSQL and NewSQL Databases for internet of things. *In Proceedings of the 2016 IEEE Bombay Section Symposium (IBSS)*. Maharashtra, India. 2014. pp. 12-14
79. Ray, P.P., Dash, D. Edge computing for Internet of Things: A survey, e-healthcare case study and future direction. *Journal of Network and Computer Applications*. 2018. pp. 1–22.
80. Singh, A. Data Migration from Relational Database to MongoDB. *Global Journal of Computer Science and Technology*. 2015. pp. 1–22.
81. Zhao, G., Huang, W., Liang, S., Tang, Y. . Modeling MongoDB with relational model. *In Proceedings of the 2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*. Washington. 2021. pp. 121

ДОДАТОК А (обов'язковий)

Лістинг програмного забезпечення мобільного додатку

```

using GMEnterprise.Services;
using GMEnterprise.Services;
using System;
using System.Net;
using System.Threading.Tasks;
using Xamarin.Essentials;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace GMEnterprise.Views
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    public partial class MainPageLogin : ContentPage
    {
        public MainPageLogin()
        {
            InitializeComponent();
            login.Text = Preferences.Get("login", "");
            password.Text = Preferences.Get("password", "");
        }

        async void OnButtonClicked(object sender, EventArgs args)
        {
            Button button = sender as Button;
            button.IsEnabled = false;
            await AuthorizationUser();
            button.IsEnabled = true;
        }

        public async Task AuthorizationUser(bool LoadOnStart = false, bool OpenAppShell = true)
        {
            App app = Application.Current as App;
            app.QueryServer = new HttpQueryServer(login.Text, password.Text);

            if (!app.QueryServer.Autoload && LoadOnStart)
                return;

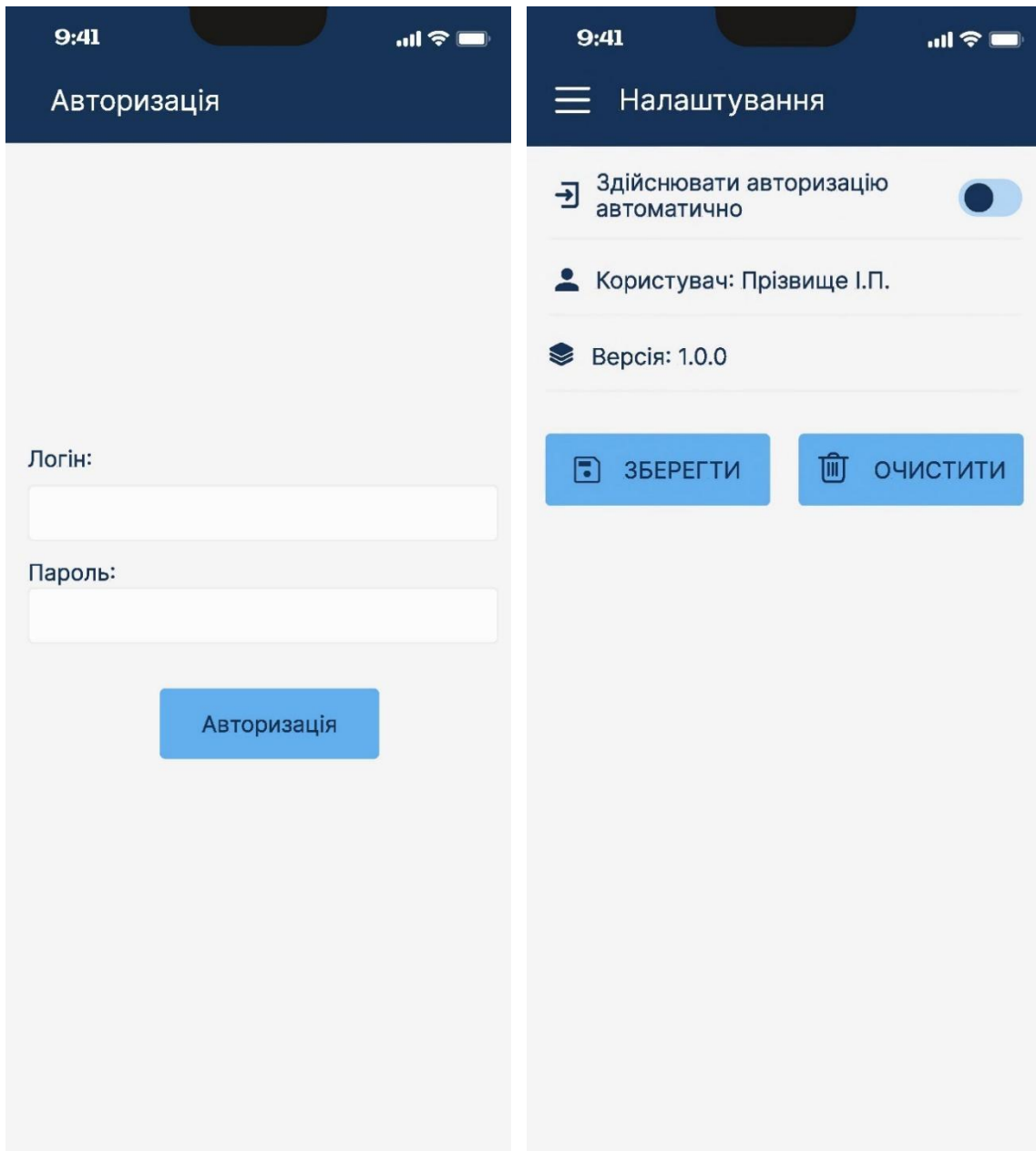
            IndicatorLogin.IsVisible = true;
            IndicatorLogin.IsRunning = true;
            if (await app.QueryServer.AuthorizationUserAsync())
            {
                if (OpenAppShell)
                {
                    app.MainPage = new AppShell();
                }
            }
            app.notificationTelegram.Collections = null;
            IndicatorLogin.IsRunning = false;
            IndicatorLogin.IsVisible = false;
        }

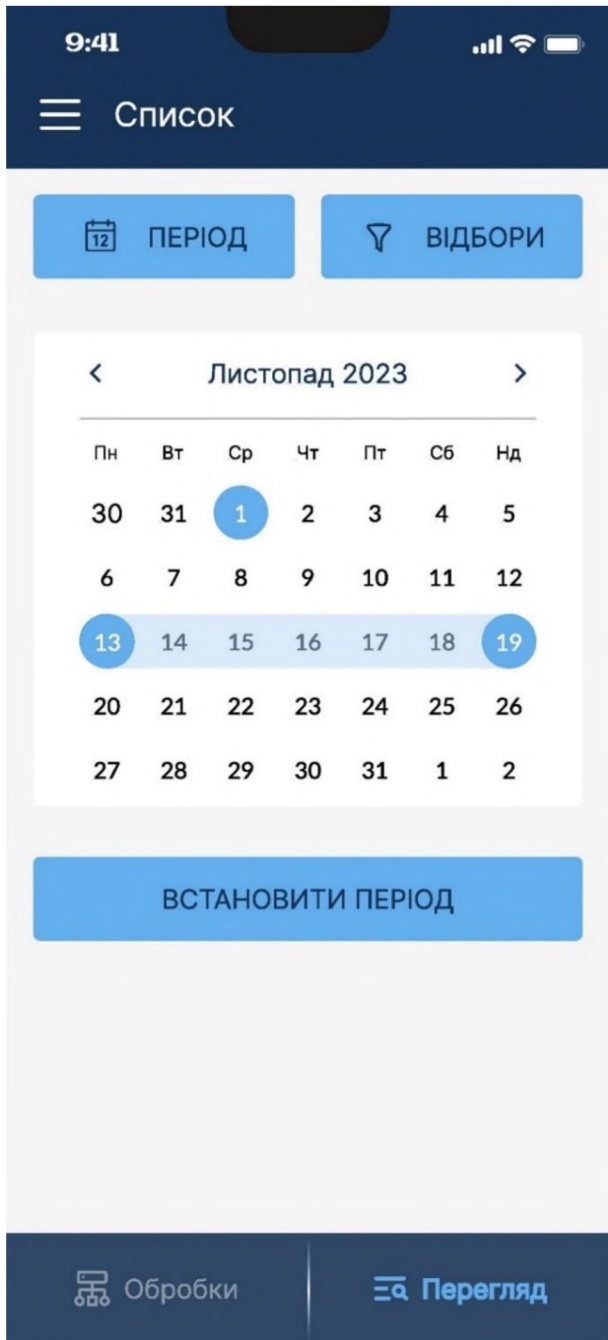
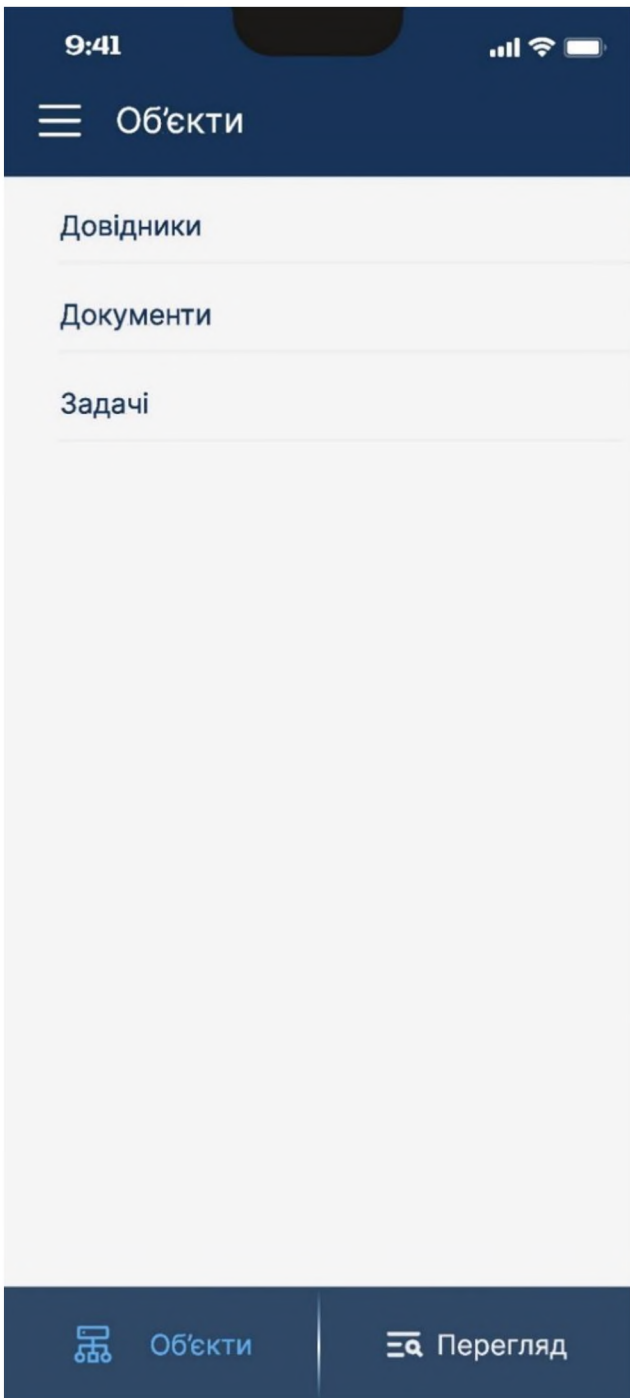
        private void ToolbarItem_Clicked(object sender, EventArgs e)
        {
            Navigation.PushAsync(new PageSetting());
        }
    }
}

```


ДОДАТОК Б (обов'язковий)


Макети мобільного додатку








9:41 📶 🔋


← Клієнт: Прізвище Ім'я



 Клієнт:
Прізвище Ім'я По батькові

 Контактна особа:
Прізвище Ім'я По батькові

 Телефон:
067 111 11 11



 Дата народження:
00.00.0000

 Email:
Email@gmail.com

 Довідники  Перегляд

9:41 📶 🔋

☰ Список



 ПЕРІОД  ВІДБОРИ

Рек 1

Рек 2

Рек 3

Поле 1	Поле 2	Поле 3	Поле 4	Поле 5.	Поле 6.
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input checked="" type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>
Знач.	Прізви... Ім'я	Прізви... Ім'я	22:36:04 22:36:04	1	<input type="checkbox"/>

 Обробки  Перегляд

ДОДАТОК В

Зображення у мобільному додатку

Зображення	Опис	Назва
	Календар	calendar
	Пошук	category
	Підтвердження	check
	Закрити	close
	Ієрархія	datamanagement
	Видалити	delete
	Довідники	directories
	Документ	document
	Документи	documents
	Таблиці	table
	Задачі	task
	Налаштування	settings
	Вибрати	select
	Зберегти	save
	Реквізити	requisites
	Звіти	report
	Виконати	form
	Редагувати	edit
	Клієнт	person

ДОДАТОК Г

Тези «Стан, досягнення та перспективи інформаційних систем і технологій».

Матеріали конференції.»

Матеріали конференції «Стан, досягнення та перспективи інформаційних систем і технологій»

УДК 004.93

МЕТОД ОПИСУ МЕТАДАНИХ ОБ'ЄКТІВ ДЛЯ МОБІЛЬНОЇ КРОСПЛАТФОРМНОЇ CRM-СИСТЕМИ НА ПЛАТФОРМІ BAF

ГЛІМБОВСЬКИЙ Б.В. (sobathka199980@gmail.com),

БОРОВИК О. В. (bov_nadpsu@ukr.net)

Хмельницький національний університет

На цей час значна кількість компаній і підприємств автоматизують бізнес-процеси і впроваджують різні спеціальні інформаційні та CRM-системи. При цьому масовості набуває використання мобільних CRM-систем, які дозволяють ефективно управляти даними про клієнтів, взаємодіяти з ними та бізнес-процесами. Для розробки мобільної крос-платформної CRM-системи на платформі BAF важливим є застосування методу опису метаданих об'єктів. Метадані є ключовою складовою такої системи, оскільки визначають структуру та властивості об'єктів, з якими працює користувач.

Сучасні мобільні технології (смартфони, планшети) відіграють ключову роль у трансформації способів ведення бізнесу та управління взаємовідносинами з клієнтами. Одним з важливих аспектів цього є використання мобільних CRM-систем, які дозволяють ефективно управляти даними про клієнтів, взаємодіяти з ними та бізнес-процесами.

На цей час існують технічні рішення, що вирішують зазначені задачі. Вони реалізовані в таких системах, як KeyCRM, SAP CRM, мобільна платформи "BAF". Але ці мобільні рішення не мають того, що потрібно бізнесу саме сьогодні - швидкодії роботи системи з великими масивами даних, безпеки даних, гнучкості в налаштуванні. Тому для розробки мобільної крос-платформної CRM-системи на платформі BAF важливим є застосування нового методу опису метаданих об'єктів. Вони є ключовою складовою такої системи, оскільки визначають структуру та властивості об'єктів, з якими працює користувач [1].

Авторський метод опису метаданих об'єктів для мобільної CRM-системи на платформі BAF характеризується такими перевагами:

1. Гнучкість та масштабованість. Метод дозволяє легко розширювати та адаптувати структуру даних відповідно до потреб бізнесу без необхідності внесення змін у програмний код.

2. Зручність і доступність. Мобільний додаток дозволяє співробітникам отримувати доступ до важливої інформації з будь-якого місця та в будь-який час, що спрощує роботу та покращує комунікацію з клієнтами.

3. Підвищення продуктивності. Завдяки системі працівники можуть ефективніше вести облік клієнтів та здійснювати операції без зайвих запитів на інформацію.

Метадані значно полегшують та прискорюють пошук потрібної інформації, обмін даними між різними системами, а також класифікацію й систематизацію цифрового контенту за певними категоріями та критеріями [2].

Визначення структури метаданих включає ключові атрибути об'єктів, такі як назва, синонім, тип, параметри вибору тощо. Ця структура має бути оптимізована для ефективної роботи мобільного застосунку та пошуку інформації в базі даних платформи BAF.

Формування метаданих відбувається для кожного об'єкта відповідно до визначеної структури у довіднику налаштувань. А їх представлення відбувається у придатному для використання форматі на мобільних пристроях. Для цього пропонується використовувати компактний та структурований формат JSON для відображення в мобільному додатку [3].

На рисунку 1 зображено макет сторінок мобільного додатку (авторизації, налаштувань, відображення списку об'єктів).

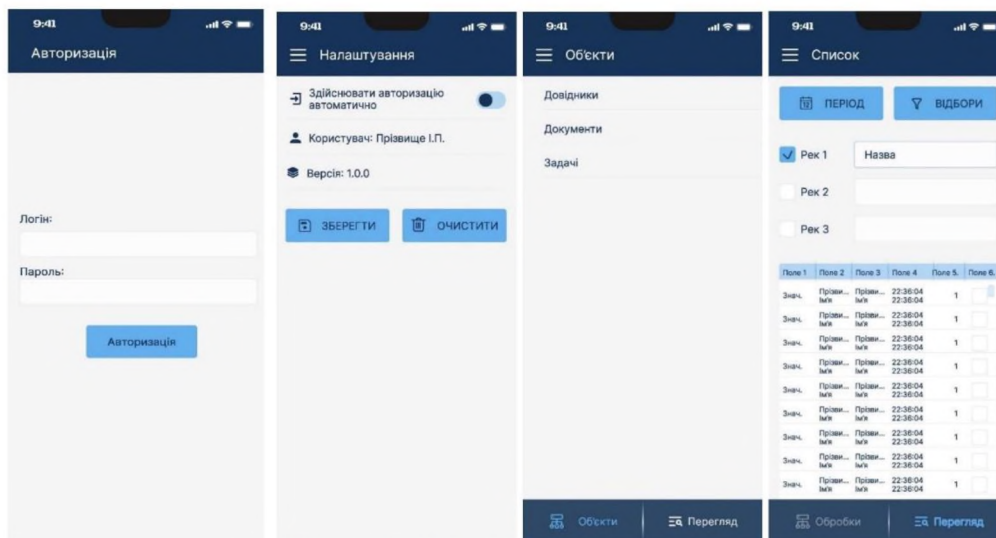


Рисунок 1 – Макет в мобільного додатку

Розроблений метод дозволяє ефективно організувати та зберігати інформацію про об'єкти, доступні на мобільному додатку. Використання запропонованого підходу сприяє покращенню доступності та зручності роботи з метаданими на мобільних пристроях.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ontotext. (n.d.). Metadata Fundamentals. Ontotext [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.ontotext.com/knowledgehub/fundamentals/metadata-fundamental/>
2. OpenDataSoft. (2021). What is Metadata and Why is it Important for Data Ontotext [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.opendatasoft.com/en/blog/what-is-metadata-and-why-is-it-important-data/>
3. Fowler, M. (2015). Metadata. IEEE Software, 22(1), 10-12. Ontotext [Електронний ресурс]. — Режим доступу до ресурсу: <https://martinfowler.com/ieeeSoftware/metadata.pdf>

ДОДАТОК Г

Презентація

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерної інженерії та інформаційних систем

Виконав ст. групи КІ2м-22-1:
Глімбовський Б.В.

Науковий керівник:
д.т.н., проф. Боровик О.В.

Метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF

Актуальність роботи

У сучасному світі велика кількість бізнес-процесів переходить на мобільні платформи. Розробка нового методу метаданих для мобільних CRM систем дозволить більш ефективно працювати з даними. Розробка мобільної CRM системи покращить взаємодію з клієнтами, забезпечити більш швидкий та ефективний доступ до інформації про клієнтів, їх історію та потреби.

Актуальність роботи

Мобільна система спростить робочі процеси та дозволить персоналу працювати з даними та клієнтами поза офісом, що призведе до підвищення продуктивності та зменшення затрат на час.

Бізнес все більше переходить до використання мобільних додатків для оптимізації робочих процесів та залучення клієнтів, що робить розробку мобільної CRM системи на платформі BAF вкрай актуальною.

Мета роботи

Дослідити та розробити ефективний метод опису метаданих об'єктів для мобільної CRM системи на платформі BAF, забезпечивши оптимальну швидкодію та розробити мобільний застосунок для роботи з методом.

Постановка задачі

Розробити метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF, дозволить полегшити та прискорити роботу з великими об'ємами даних в системі.

Постановка задачі

Розробити метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF, дозволить полегшити та прискорити роботу з великими об'ємами даних в системі.

Об'єкт та предмет дослідження

Об'єкт дослідження – CRM система, інформаційні технології, моделі опису метаданих об'єктів у CRM-системах.

Предмет дослідження - метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF.

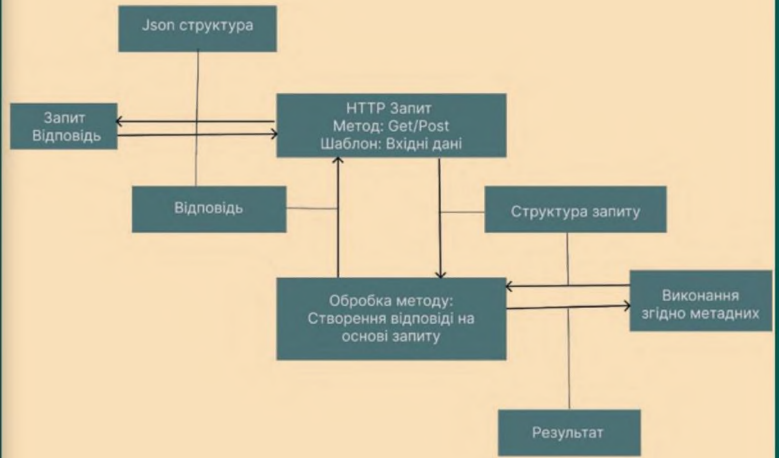
Наукова новизна

Розробка нового методу опису метаданих об'єктів, який сприяє більш ефективній роботі з даними, та створенні мобільного додатку, який реалізує зазначений метод. Крім того, дослідження в області оптимізації використання HTTP сервісів і протоколу OData, а також вивчення аспектів безпеки використання HTTP сервісів у мобільних CRM системах, дозволяють отримати нові знання про забезпечення безпеки обміну даними в таких системах.

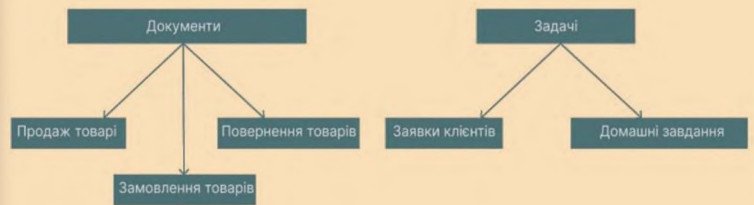
Публікації за матеріалами магістерської роботи

За темою дипломної роботи опубліковані тези у матеріалах наукової конференції «СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ - 2024» (18-19 квітня 2024 р.).

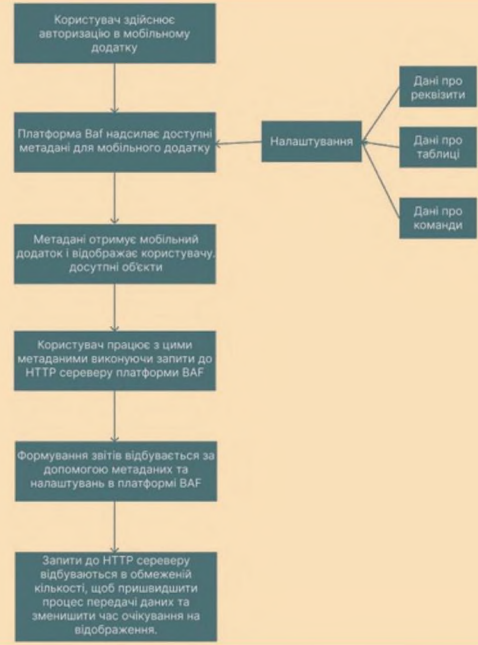
Блок схема роботи HTTP сервісів платформи BAF



Ієрархічної структура довідника налаштувань



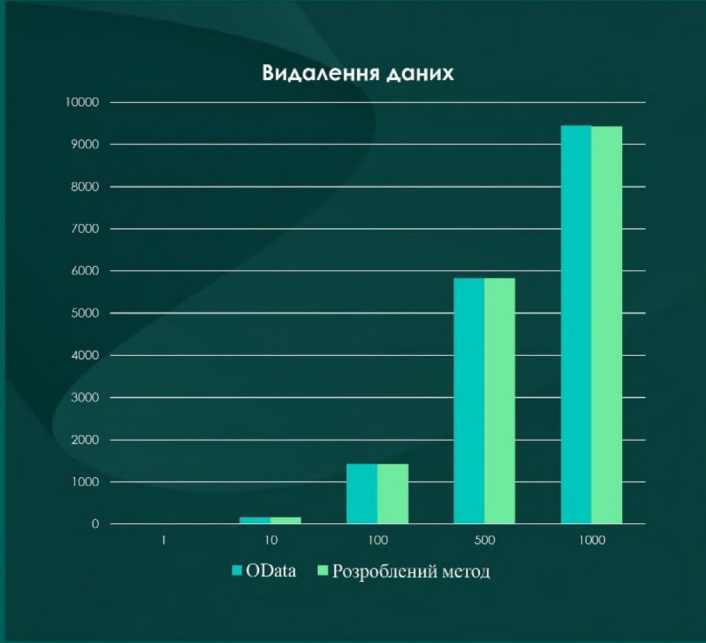
Структура роботи методу



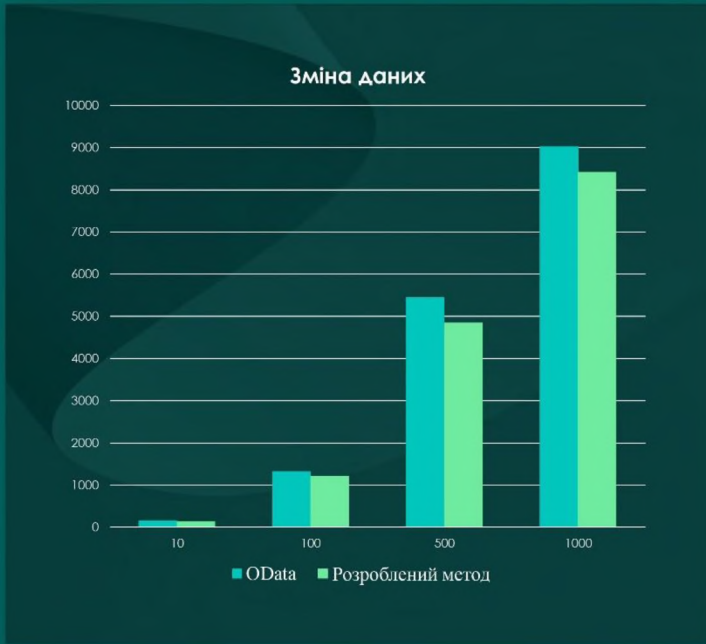
Читання даних



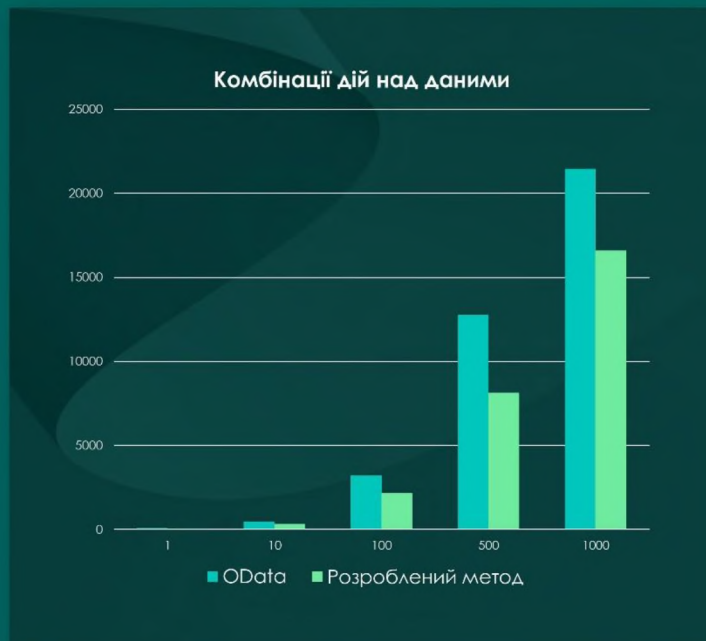
Оцінка швидкодії запропонованого методу при читанні даних



Оцінка швидкодії запропонованого методу при видаленні даних



Оцінка швидкодії запропонованого методу при зміні даних



Оцінка швидкодії запропонованого методу при комбінації дій над даними

Висновки

Перспективами подальших досліджень у даній роботі є застосування розробленого методу опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF до комплексного вирішення задач системного аналізу, в яких розпізнавання та запобігання позаштатним, критичним і катастрофічним ситуаціям у складній системі є складовою частиною задачі. Зокрема, запропонований метод дозволяє:

1. Підвищити ефективність управління взаємовідносинами з клієнтами шляхом оптимізації бізнес-процесів та забезпечення гнучкості у конфігурації CRM-системи.
2. Підвищити продуктивність роботи користувачів мобільної CRM-системи на платформі BAF завдяки ефективному управлінню метаданими.
3. Підвищити рівень безпеки даних та прозорість управління доступом до застосунку за рахунок впровадження механізмів забезпечення безпеки та контролю доступу.

Дякую за увагу!

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016265522

Дата перевірки:
20.05.2024 17:42:34 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
20.05.2024 19:23:04 EEST

ID користувача:
100005591

Назва документа: Глімбовський_Метод опису метаданих об'єктів для мобільної кросплатформної CRM-систе...

Кількість сторінок: 93 Кількість слів: 13454 Кількість символів: 105491 Розмір файлу: 3.21 MB ID файлу: 1016055453

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

11% Схожість

Найбільша схожість: 4.09% з Інтернет-джерелом (<https://ela.kpi.ua/handle/123456789/31906>)

10.8% Джерела з Інтернету

197

Сторінка 95

1.26% Джерела з Бібліотеки

57

Сторінка 97

0% Цитат

Не знайдено жодних цитат

Посилання

1

Сторінка 97

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

52

Підозріле форматування

25
сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 12%**

ID: 126716 Назва: МКР Метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі BAF Додано в БД: 2024-05-20 Автора: Б.В. Глімбовський Керівники: О.В.Боровик Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	85320	809	856 (1%)	13 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Глімбовський Богдан Валентинович

Тема: Метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі VAF

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень _2_ Кількість сторінок записки _81_

1. Короткий зміст роботи та прийнятих рішень: Метою магістерської роботи є дослідити та розробити ефективний метод опису метаданих об'єктів для мобільної CRM системи на платформі VAF, забезпечивши оптимальну швидкодію та розробити мобільний застосунок для роботи з методом.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи:

У першому розділі проведено аналіз предметної області систем управління відносинами з клієнтами (CRM), визначено актуальні проблеми розробки та впровадження мобільних CRM-додатків.

У другому розділі для ефективного управління метаданими в сучасних інформаційних системах запропоновано використання апарату алгебри каскадів. Це дає змогу формалізувати опис структури методу опису метаданих, ввести необхідні операції над каскадами, а також розробити систему показників для всебічної оцінки ефективності функціонування методу.

У третьому розділі наведено розроблений метод опису метаданих об'єктів, який передбачає використання ієрархічного довідника для централізованого управління налаштуваннями. Розглянуто структуру довідників, що зберігають інформацію про поля, таблиці, команди тощо. Запропоновано механізми

забезпечення безпеки та контролю доступу, зокрема ведення журналу авторизації користувачів та блокування об'єктів.

У четвертому розділі описано процес тестування та налаштування системи, а також здійснено ґрунтовне порівняння ефективності розробленого методу опису метаданих з існуючим протоколом OData. Результати тестування, представлені у вигляді таблиць та діаграм, демонструють значне підвищення продуктивності при використанні розробленого методу, особливо при роботі з великими обсягами даних.

4. Позитивні сторони роботи: інноваційність, підвищення ефективності, гнучкість та масштабованість _____

5. Негативні сторони роботи: вартість впровадження, складність адаптації _____

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: добре (4/С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Бедратюк Деонід петрович, професор, зав. Кафедри ІІЗ, ХНУ

“21” травня 2024 р.

_____ (підпис)

Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорущенко Т. О.

Глімбовського Богдана Валентинович

ПІБ здобувача вищої освіти

ФТ, 2 курсу, групи КІ2М-22-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2024 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод опису метаданих об'єктів для мобільної кросплатформної CRM-системи на платформі VAF

Автор: Глімбовський Богдан Валентинович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Боровик О. В., д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості Unicheck, складає 11% і адресується до 81 першоджерела; та системою Anti-Plagiarism складає 0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



О. В. Боровик

Гарант ОП



О. С. Савенко

Завідувач кафедри КІС



Т. О. Говорущенко